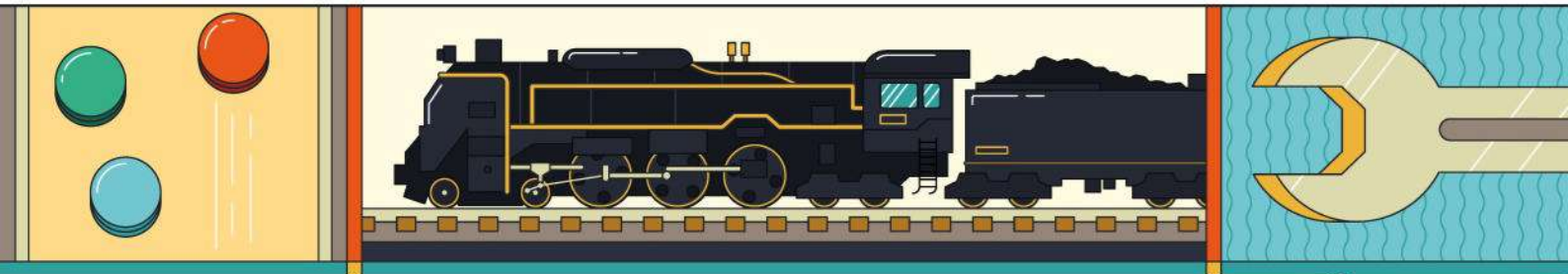
MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | April 2023 | Issue #65

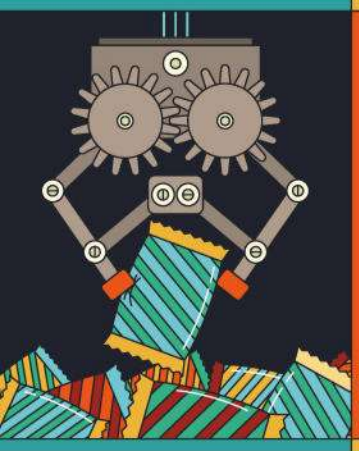


Bluetooth

Wirelessly connect to Pico W

DIY GAMES

16 maker projects to keep boredom at bay

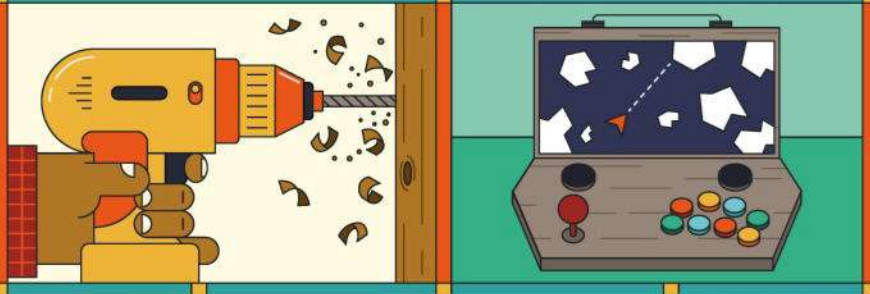






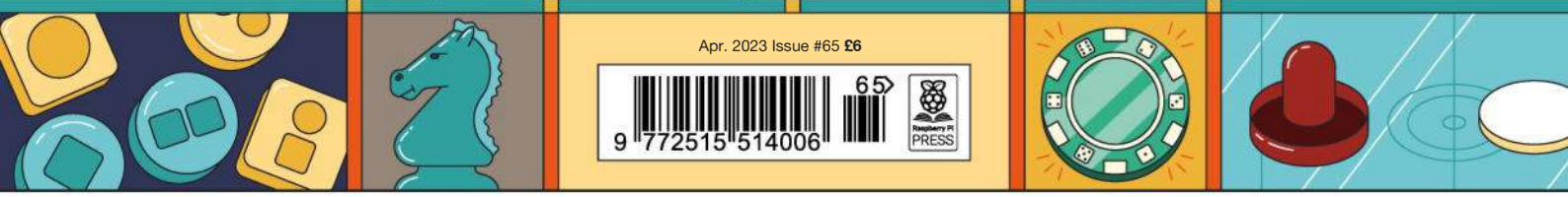
Soldering

Working with surface-mount



Flexible 3D Prints

Wibbly-wobbly prints with TPU



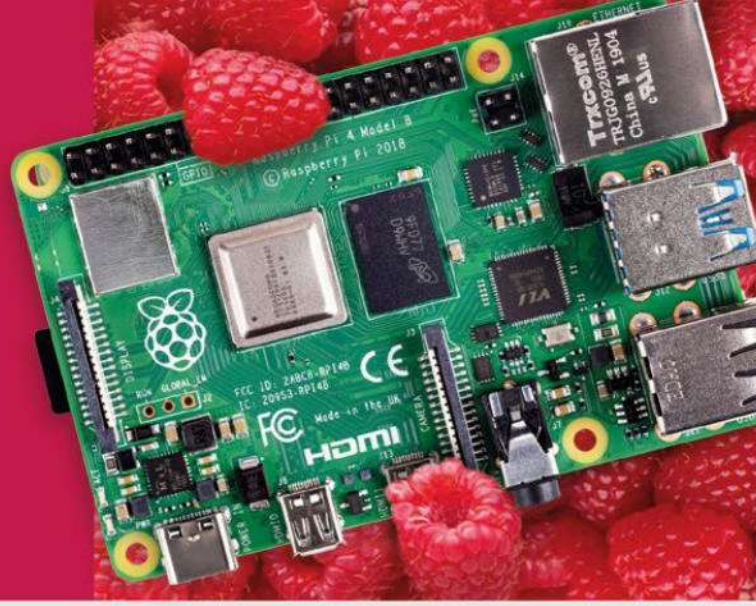
Apr. 2023 Issue #65 £6

9 772515 514006 65

Raspberry Pi PRESS

FLIGHT **SCORPIO** STAMP **LIGHTS**

American **Raspberry Pi** Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:

adafruit · sparkfun. ELECTRONICS · PIMORONI · HiFiBerry · and many others!

Price, service, design, and logistics support for **VOLUME PROJECTS**



USA **PiShop.us**

Canada **PiShop.ca**

Raspberry Pi APPROVED RESELLER



Welcome to HackSpace magazine

Games might seem frivolous, but they play an important role in society. As children, we learn how to navigate the complex world of interpersonal dynamics in the playground. We learn how to lead, how to be a part of a team, how to win, and how to lose. Games provide us with a safe space to experiment and develop.

As adults, games can still provide us with a great learning experience, not least because they give us a chance to flex our 'making' muscles. That could be trying a new technique, investigating a new material, or working with a new tool.

If things go well, then you've got a great new amusement to enjoy. If things don't, hopefully you've learned a few things in the process. Either way, games give us a great space to continue to learn and develop our skills.

As adults, games can still provide us with a great learning experience, not least because they give **us a chance to flex our 'making' muscles**

BEN EVERARD
Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play



PAGE 30
FREE PICO W
WHEN YOU SUBSCRIBE

EDITORIAL
Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN
Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell

Photography
Brian O'Halloran

CONTRIBUTORS
Marc de Vinck, Nicola King, Carl Nicholson, Jo Hinchliffe, Phil King

PUBLISHING
Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

recycle

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



16

06 SPARK

06 **Top Projects**
Amazing builds to dazzle and inspire

16 **Objet 3d'art**
The guitar Leo Fender wishes he could have made

18 **Letters**
What's pushing buttons in the makersphere

21 LENS

22 **Toys and Games**
Playful projects to relax the mind

32 **How I Made:** Greening the Spark
A homemade electrical power grid simulator

38 **Interview:** Kitronik
Building gear for the next generation of makers

46 **In the workshop**
Blinky lights inspired by Islamic art

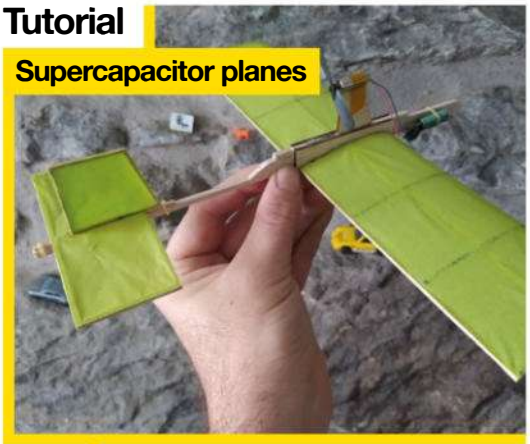
Cover Feature



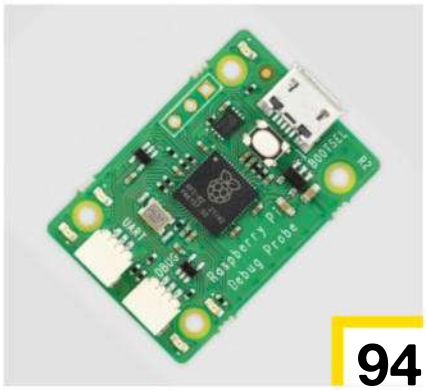
22

Tutorial

Supercapacitor planes



64 Who knew the humble capacitor could power a flying machine?



94

CONTENTS



06



84

Review

Spintronics



92 Model electronics with various gears, chains, and buttons

Interview

Kitronik



38 From Nottingham to the world – maker gear for the inventors of tomorrow



90

49 FORGE

50 **SoM** TPU
Discover TPU: flexible 3D printer filament

52 **Tutorial** Surface-mount soldering
Electronic connections in an itty-bitty space

58 **Tutorial** Raspberry Pi
Track the tides with an e-ink display

64 **Tutorial** Powered flight
Build a supercapacitor-powered plane

72 **Tutorial** Metal stamping
Hammer, meet metal. Metal, meet design

76 **Tutorial** Raspberry Pi Pico
Debugging without wires

78 **Tutorial** 3D printing
Print faster with a speedy slicer

83 FIELD TEST

84 **Best of Breed** Robot pets
Landlord won't let you have a dog? Build a robotic one instead!

90 **Review** Adafruit Feather RP2040 Scorpio
Drive many, many NeoPixels from a Feather board

92 **Review** Spintronics
Understand electronics via gears and chains

94 **Review** Raspberry Pi Pico Debug Probe
Plug this in and fix all your problems. Well, some of them

96 **Crowdfunding**
Harness wind-power, save polar bears

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits. HackSpace is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to HackSpace, c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

REGULAR

3D printing meets stoneware

By **Allie Katz** katzcreates.com

3D printing is great for all sorts of reasons, and one oft-overlooked application is on 3D-printing moulds. This stoneware vase by Allie Katz looks like a traditional glazed, fired piece of pottery but it began life as a CAD design. Allie printed it, then turned it into a mould to remake the design in a more traditional material.

Allie's working on a video about the technique and how people can potentially create their own. Keep an eye out for it on their YouTube channel: **youtube.com/@katzcreates**.



Right
Stoneware clay shrinks by close to 30% on firing, so this vase is a good bit smaller than the original 3D-printed version




KONG

Model railway advertising display

By Christoph Ketzler

hsmag.cc/ModellbahnWerbedisplay

The world of model railways is full of creativity. This is the latest idea we're seen to come out the model railway world, perfectly executed by Christoph Ketzler. He's used tiny screens (0.96-inch (80×160px) or 1.14-inch (135×240px)) as advertising hoardings in his model transport systems, and connected them via WLAN so that the user can upload their own PNG advertising images.



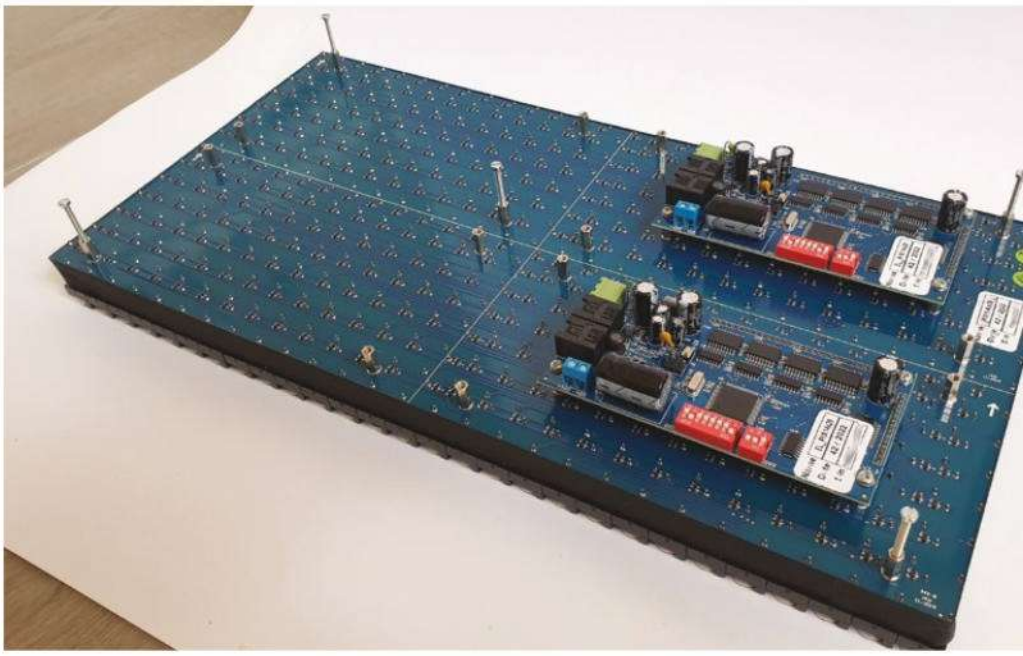
Left
Even model commuters must be told to consume!

Flipboard Tetris

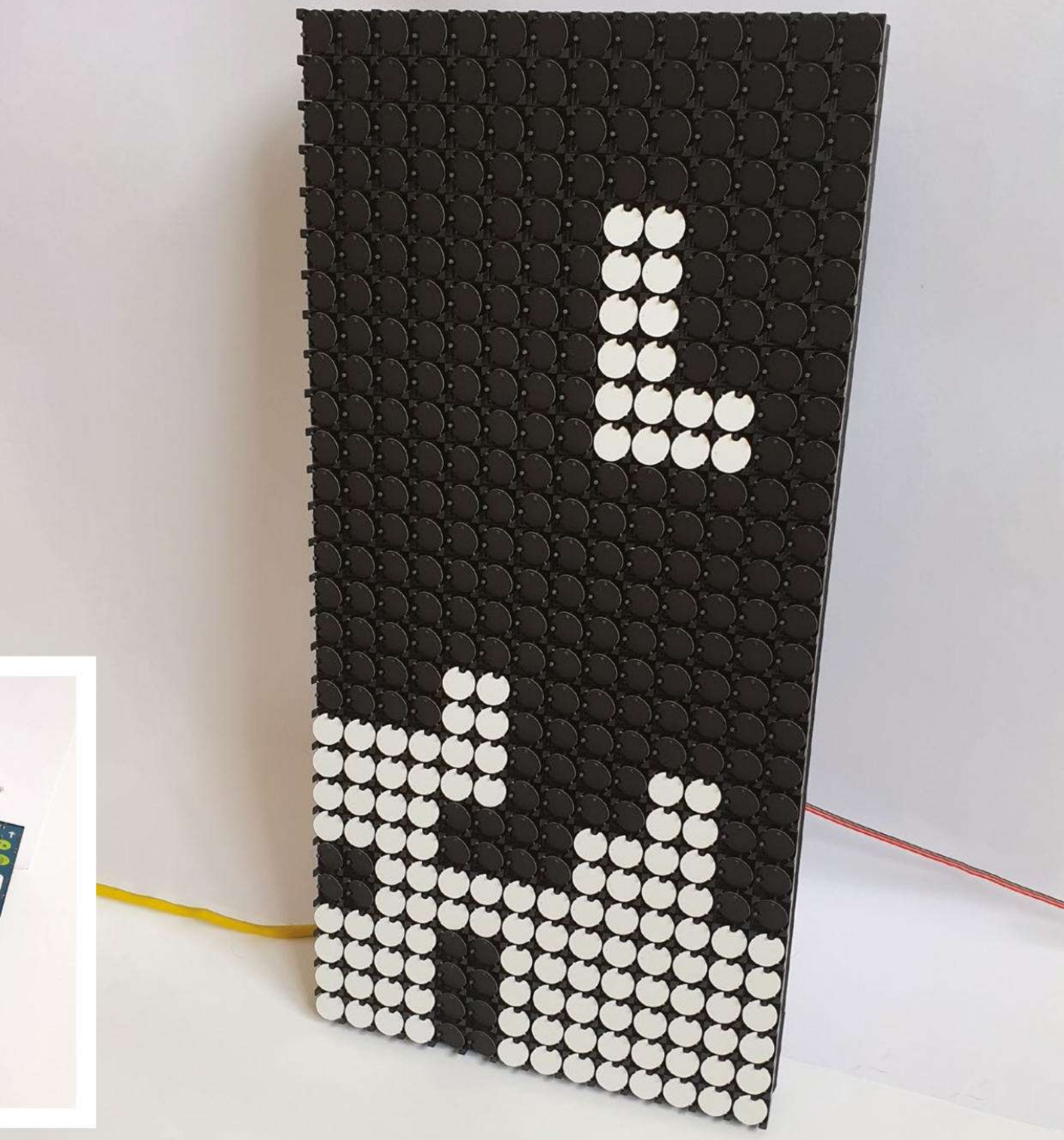
By **Owen McAteer** hsmag.cc/FlipTetris

We all like Tetris – there's even a feature film coming out soon about it, so if there's anyone out there who still hasn't heard of it, they will soon. However, this is the first time we've seen Tetris implemented on a flip-dot panel.

The visionary behind this work is Owen McAteer, aka Motus Art. His flip-dot Tetris is fully playable, and it makes a wonderful click-clack noise as the solenoids built into the screen make the dots flip from black to white. Like all the best stuff, you can download Owen's code from his GitHub page.



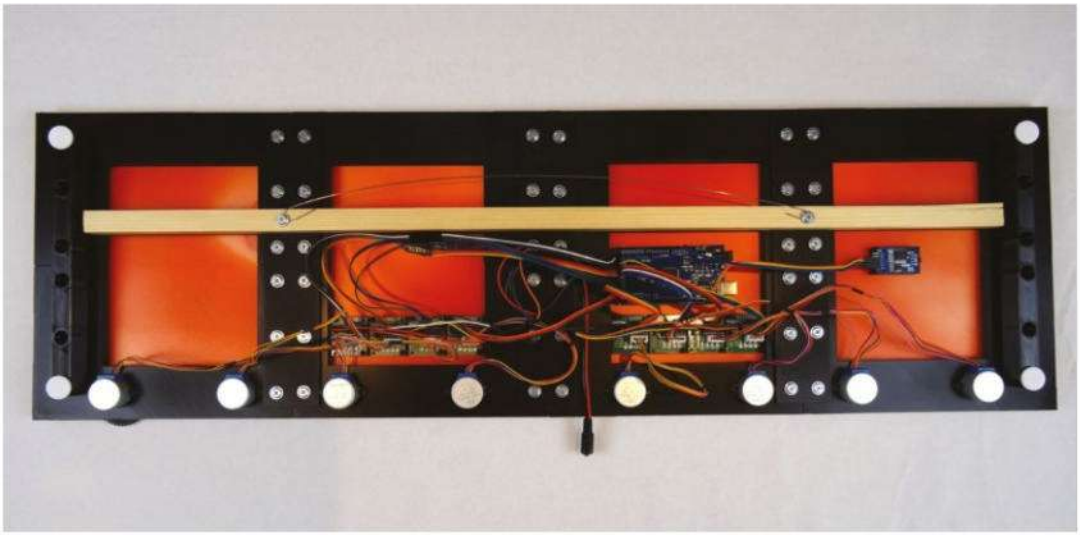
Right
The panel comes from flipdots.com, in Poland



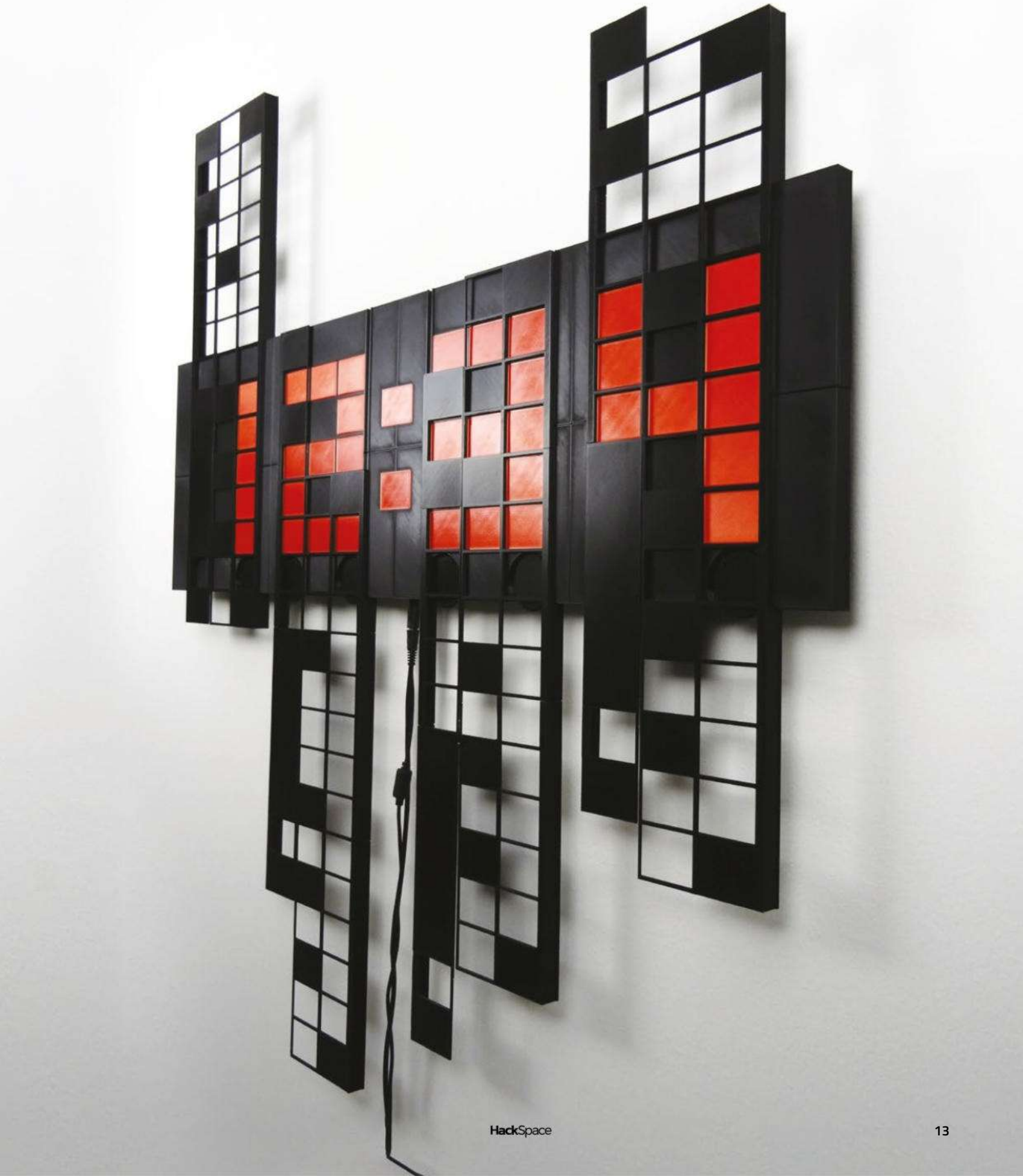
Time slider

By Hans Andersson tiltedtwister.com

Last issue, we brought you a clock with digits that slid up and down to display the time; this issue, we've discovered a leap forward from this, with grids of black and white squares sliding up and down over a red light source to make digits. It's powered by an Arduino Mega that controls eight small stepper motors, and there's a real-time clock module (DS3231) to keep track of time.



Right
We've lost far more time than is healthy watching the video of this wonderful clock



Shortlife

By **Dries Depoorter** driesdepoorter.be/shortlife

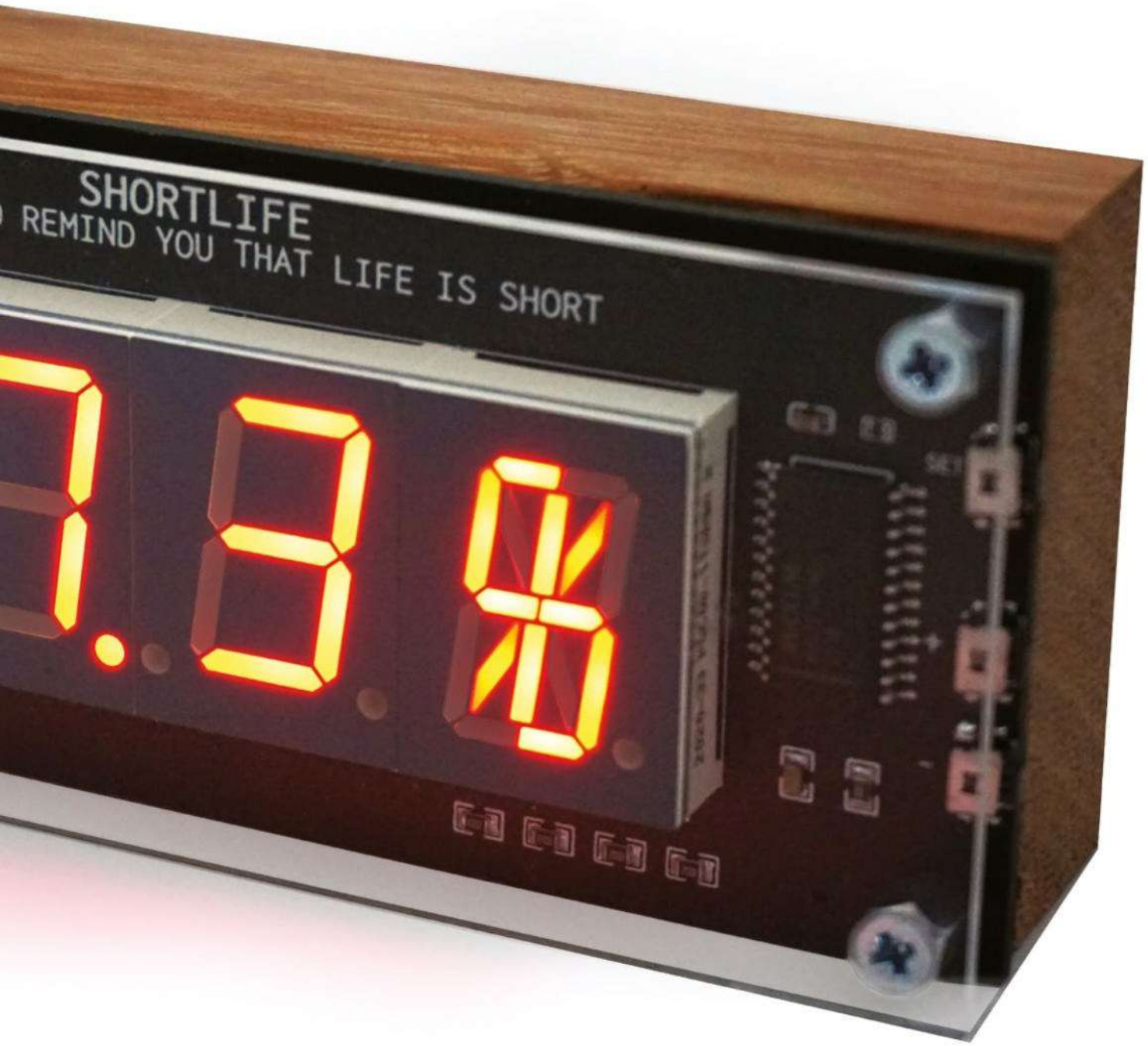
Life is short! Why not buy a clock that counts down until your expected death, paralysing you with fear and indecision as you watch what's left of your time on Earth drain away? Or you could see the countdown as a warning and use it to motivate you to get out and do the things you've always wanted to do. It's up to you.

When you buy the Shortlife clock, you supply your date of birth, country of residence, and sex; from that, Dries will calculate your estimated life expectancy based on data from the World Health Organization (WHO); the user just needs to add the time and date. We like the idea of this; it's a little like having a voice on your shoulder saying, "remember you are mortal".





Above
The clock will be delivered including an USB adapter (1A), micro USB cable, and manual


SHORTLIFE
REMIND YOU THAT LIFE IS SHORT

Objet 3d'art

3D-printed artwork to bring more beauty into your life



In issue 55 of HackSpace mag, we printed a ¾ size guitar. It failed utterly; we assumed that the neck was mostly under compression, and so printed it so that the layer lines were perpendicular to the strings, rather than parallel. It broke within minutes.

With that recent failure still burning, we could only look on with envy at Mikoláš Zuza's amazing Prusacaster. It's a full-sized guitar that uses the cannibalised parts of a Telecaster-style guitar kit (available for a bargain £79/€89 from **hsmag.cc/HarleyBentonT-Style**), as well as a 3D-printed body and pick guard.

The section of the body that the bridge and the neck attach to has to be strong enough to cope with approximately 50 kg of force running through the strings. The default PrusaSlicer settings use just two parameters, but this wasn't strong enough to stop the body deforming under stress; Mikoláš increased this value to seven parameters, and reprinted the part with 25% cubic infill, and it's been fine since. The joins between each part have large flat surfaces, ideal for sticking together with superglue.

Want to print your own and, in the process, learn about guitar construction? Go to **hsmag.cc/Prusacaster** and get stuck in!



Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction), let us know at **hsmag.cc/hello**

LASER CUTTER

I've been kitting out my maker space (read: shed) for a while. Now that I'm retired, I actually have time to use the various tools and things I've accumulated over the years. Your article on laser-cut boxes made me wonder if I should add one of these to my shed. They seem very affordable now.

Mark
Doncaster



Ben says: We're very wary of the super-low-cost laser cutters on the market. They generally have no enclosures, no fume extraction, and no air assist. Depending on what way they fail, they can poison you, blind you, or burn down the house and really aren't to be messed with. Actual safe systems can be expensive, though. There are plenty of companies that will laser-cut parts for you at reasonable prices, or you can even do what people did in ye olden times – print out a template on a regular printer and cut around it.



INKY FRAME

I love the idea of a photo frame that keeps updating, but don't like the look of LCD screens. Thanks for reviewing the Inky Frame from Pimoroni. Looks like it might be just what I'm after.

Sally
Carlisle

Ben says: e-ink has some weaknesses – particularly around update speed – but there really is no other technology like it for images that have to look good under natural light. You don't get super-duper high-res quality, but they also don't glow on the wall.

SOLDERING IRONS

I've been soldering occasionally for a few years and struggle quite a bit. The solder just doesn't seem to work for me the same way it does in 'how to solder' videos. After reading your cover feature in issue 64, I wonder if I should get a new soldering iron.

Paul
Woking

Ben says: Solder is funny stuff. It reacts differently depending on how hot it is and how hot the things around it are. With practice, you can get a pretty good joint out of most soldering irons, but when you're just getting started and don't know how to read the signs, it can be difficult. It can be hard to quantify the differences between soldering irons, but some just work much better than others. If the one you've got isn't working for you, it could be worth considering a new one. If there's a hackspace or maker space local to you, you might be able to try out some other options without spending too much money.



TRUST
STARTS HERE



From genuine, manufacturer-warrantied components to millions of in-stock parts shipped same day, be confident Digi-Key will get you what you need—when you need it.

Visit digikey.co.uk today, or call 0800 587 0991.

Digi-Key ELECTRONICS

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2023 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 32

HOW I MADE:
GREENING THE SPARK
A simulated national power grid controlled by 10,000 lines of Python and a Raspberry Pi

PG 38

INTERVIEW:
KITRONIK
Teacher- and child-friendly maker gear for the next generation of inventors

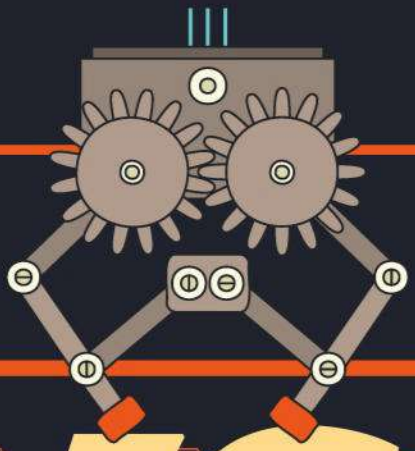
PG 46

IN THE WORKSHOP
LEDs wrangled into the shape of an Islamic-style geometric pattern

PG 22

TOYS & GAMES
Build your own maker-friendly DIY diversions







TOYS & GAMES

It's all fun and games with this compendium of playful projects.

Phil King digs out his building blocks

Many projects are fun to build, but making your own electronics-based game or toy means you can continue to enjoy playing with it afterwards. We've rounded up a wide array of fun-filled projects to inspire you to get building.

You could even create your own mini amusement arcade with a pinball table, air hockey, claw machine, coin dozer, and multiplayer whack-a-mole game – useful for taking out your frustration on if a project isn't working right! Not to mention adding a mini arcade cabinet or something more esoteric like a Dutch shuffleboard with built-in beer dispenser.

Alternatively, have fun making an automated chessboard, a plastic tags game, or an interactive TTRPG board – and spice up your tabletop gaming with accessories like electronic dice and light-up poker chips. Or why not build an IoT LEGO world or a Nerf-gun-equipped robot? With a little imagination, it's all to play for.





AMUSEMENT ARCADE

Recreate all the fun of the arcade without needing a pile of coins to play

Project

MINI PINBALL MACHINE

hsmag.cc/MiniPinball

After abandoning an attempt to build an electromechanical pinball table, maker Chris Dalke set about creating a desktop one with a 7-inch HDMI touchscreen instead. "I wanted to retain the feeling of a physical arcade game with intense sound, vibration, and colours, as well as the tactile response of the inputs," he says.

Powered by a Raspberry Pi 4, his screen-based version of pinball is coded in C++ and OpenGL, using the open-source software development library raylib. An Arduino is used for the button inputs and LED matrix scoreboard. The enclosure is laser-cut from maple wood.

If your carpentry and mechanical skills are up to it, you could have a go at building a full-size physical pinball table, like Frogger1108's beautifully finished Arduino-controlled project: **hsmag.cc/ArduinoPinball**.





Project

CLAW MACHINE

hsmag.cc/ClawMachine

Hate how those arcade claw machines always seem to drop the prize just before it reaches the exit chute? Then why not build your own version? That's just what Ryan Bates did, using an Arduino Mega and custom PCB for the electronic brains.

Unlike traditional arcade claw machines, it gives you full control over the claw in the X, Y, and Z axes, along with the claw grip itself, for a more skilful game. The only restriction is time.

"Everything on the machine is custom-designed," says Ryan, "from the XYZ gantry, to the claw, to the game logic. Stepper motors move the gantry, and a servo motor controls the claw (giving the claw an analogue grip, not just open/close)."

While it's not available as one of Ryan's DIY arcade kits, the claw machine's CAD and 3D print files and Arduino code can be found on his website: **retrobuiltgames.com**.

Project

MULTIPLAYER WHACK-A-MOLE

hsmag.cc/WhackAMole



The concept of bashing fake pop-up moles on the head with a rubber mallet dates back to 1975, when the original Mogura Taiji ('mole extermination') game was released in Japanese arcades.

Liam Eggermont's version replaces the moles with light-up hemispheres on springs mounted in a wooden enclosure. A Hall effect sensor under each hemisphere detects the latter's descending magnet when it has been bashed, but you only score a point when it's lit.

A Raspberry Pi is used to read the sensors via three ADCs, while an Arduino controls the LED strip. Liam designed three custom boards to connect everything up. Follow the link above for build instructions and the custom libraries needed.

Project

AIR HOCKEY ROBOT

hsmag.cc/AirHockeyRobot



This amazing project by Ondřej Sláma and Dominik Jašek, for their college course thesis, involved creating robot movement control algorithms, computer vision, game strategy algorithms, and a user interface.

Designed in Fusion 360, the plywood table features a mesh of 920 holes to create an air cushion for the puck to float on. An 'H-bot' design is employed to move the AI player's paddle; held in a 3D-printed housing, the paddle is moved from side to side using a pulley and belt system with two stepper motors. A Raspberry Pi 4 is connected to a Camera Module mounted on an overhead frame to track the puck's position.

If this ambitious project sounds too daunting to recreate, you could always build a standard two-player air hockey table such as this Arduino-based one: **hsmag.cc/ArduinoAirHockey**.

Project



HEVERLEE SJOELEN SHUFFLEBOARD

hsmag.cc/Sjoelen

The Dutch version of shuffleboard, Sjoelen, involves sliding wooden pucks down a long table into a set of four scoring boxes.

Commissioned to produce a special version of the game by beer brand Heverlee, Grant Gibson simplified the maths-heavy scoring system and employed a Raspberry Pi to dispense prizes (cans of beer) to winners. He also equipped it with a camera to capture players' reactions.

Infrared sensors are used to detect when the puck passes through the scoring gates. Cooled by a custom Peltier heat exchanger and old PC fans, the drinks dispenser uses car central locking motors to operate a see-saw mechanism to serve the cans.

Project

VECTREX MINI

hsmag.cc/VectrexMini



Launched in 1982, the Vectrex games system soon disappeared, which is why it's now a rare collector's item. Its curious design incorporated a vertically oriented vector CRT monochrome monitor – colour could be added with the addition of an acrylic screen overlay that came with each cartridge-based game.

Following the recent trend for mini-consoles, Aussie retro games fan Brendan Meharry decided to create his own miniature Vectrex. "I thought it would be fun to 'minify' something more obscure, especially since the original included a built-in monitor," he explains.

For the latter he used a 2.5-inch LCD screen, fitting into a 3D-printed enclosure. The Blender models for that and the separate controller unit can be found at **hsmag.cc/VectrexBlender**.

The games run on a Raspberry Pi running a Vectrex emulator in RetroPie, while audio is supplied via a super-small PAM8302 amplifier.

Project

TABLETOP ARCADE UNIT

hsmag.cc/TableTopArcade



Building an arcade cabinet is a popular project, but this tabletop example stands out from the crowd with its genuine walnut wood enclosure and head-to-head multiplayer setup – ideal for staring into the eyes of your opponent as you beat them at Street Fighter II.

Reddit user mw33121 used wood offcuts from other projects (including an impressive walnut laptop) to construct the cabinet. He made it for his brother after the latter saw the first one he built and dropped some hints!

The cabinet is equipped with high-quality Sanwa arcade buttons and joysticks. Two 14-inch laptop panels were repurposed for the screens, using LCD driver boards, with an HDMI splitter to mirror the video output from a Raspberry Pi running RetroPie for retro games emulation.

Project



ARDUINO COIN DOZER

hsmag.cc/CoinDozer

No amusement arcade would be complete without a coin dozer/pusher machine, like the one in the *Tipping Point* TV game show. Feed your coins into the slot and cross your fingers that they push others off the ledge as it moves back and forth.

This homemade coin dozer is another creation of Ryan Bates of Retro Built Games. It was formerly available as a kit, but the build instructions and Arduino code can still be found on his website.

The machine is constructed from a stack of laser-cut pieces. An Arduino-controlled servo or stepper motor is attached to a gear mechanism to move the pusher shelf – alternatively, you could make a hand-cranked version. There's even a tilt sensor to detect if someone is trying to cheat by bumping the machine. →

BOARD & TABLETOP GAMES

There's no time to be bored while making and then playing games like these

Project

AUTOMATED CHESSBOARD

hsmag.cc/AutomatedChessBoard

This automated chessboard by Greg is a particularly fine example. The super-sleek build sees the AI pieces sliding magically across the board. Knights cleverly sneak through gaps to bypass other pieces, while any taken pieces are slid to the edge of the board.

The AI pieces are propelled by a strong electromagnet hidden in the base of the machine, moved along two rails (the X and Y axes) by stepper motors – as inspired by 3D printer mechanisms. An LCD screen and two arcade buttons are used as a chess clock (you can also play against another human).

Most surprising is that rather than a single-board computer, it's all controlled by a humble Arduino Nano. For the AI, it runs a port of the Micro-Max chess engine which weighs in at less than 2kB.

Helpfully, the maker has provided meticulously detailed build instructions, 3D print files, and Arduino code, available via the link above.





Project

GIANT BATTLESHIPS

hsmag.cc/Battleships

While you could play the classic game of Battleships with pencil and paper, or the plastic peg board game, Dan Aldred's oversize version is a feast for the senses. The 'board' comprises a giant 10×10 NeoPixel matrix, which Dan originally built for disco lighting, with spray-painted vitamin bottles cleverly used as light diffusers.

Rather than typing in co-ordinates to target their torpedoes, the player uses a 1960s Bakelite telephone to dial them in. While Dan originally intended to use a single Raspberry Pi for the project, he ended up using two: one for the NeoPixel matrix and another for the phone.

As you turn the dial, a circuit with a GPIO pin is interrupted by the relevant number of clicks. Each co-ordinate is sent in turn, using Python sockets, to the LED board's Raspberry Pi. The phone's handset also gives voice instructions and plays sound effects such as explosions.

Project

INTERACTIVE TTRPG TABLE

hsmag.cc/TTRPG



Tabletop RPGs are enjoying a resurgence in popularity, but one of the most difficult parts of playing one can be juggling different game maps. So you may want to consider upgrading your GM's table with an old TV and a Raspberry Pi. This is just what The MagPi's features editor Rob Zwetsloot did for this project, whose build process is detailed in issue 86 of that magazine (**magpi.cc/86**).

For the interactive part, this wooden table houses a TV panel showing a full-screen web browser (on a Raspberry Pi) that displays the current game map. The browser is pointed to Roll20 (**roll20.net**), an online service that enables communication between TTRPG players for remote games, but can be used to show game maps for local players you invite round.

Project

TAG JOUSTS

hsmag.cc/issue56

The best portable games don't need a board to play. This one comprises a set of 3D-printed tags, which Rob Miles explained how to create from Python code macros within FreeCAD (**hsmag.cc/FreeCAD**) in his HackSpace issue 56 tutorial.

Each player starts with 16 tags of one colour. In each round, they get to challenge the opponent(s) to a 'joust'. This begins with rolling the six-sided die to determine which shape will be used for scoring in that round: squares, circles or, if it's a blank face, whichever the attacking player decides. Players then pick a tag to use and reveal the underside to count the relevant shapes; whoever has the most will win the joust.

Detailed instructions, 3D print files, and example Python macros are available from Rob's GitHub repo: **tagjousts.xyz**.



Project

QUIZ BUZZERS

hsmag.cc/QuizBuzzers

There's nothing quite like a good quiz to rouse the competitive spirit in assembled friends and family. To make it feel more like a TV game show, why not add some light-up buzzers for players to push?

A set of properly interconnected buzzers – so that only the first one pressed lights up and locks out the others – can be expensive to buy. Not to fear, because in The MagPi issue 115 (**magpi.cc/115**), Rob Zwetsloot's step-by-step tutorial shows you how to make your own set of smart buzzers using arcade buttons and 3D-printed boxes.

With a Raspberry Pi running a Python program, you can discover in which order the buzzers were pressed and even program them to play custom sounds. →





Project

LIGHT-UP POKER CHIP

hsmag.cc/LEDPokerChip

If your regular poker game is no longer exciting enough, you could spice it up with some of these blinking LED chips. Each one is actually a circular PCB of the same size as a regular chip. Maker Alex Wulff has made the design available via OSH Park, so you can just order the bare PCBs, onto which you'll need to solder components including five LEDs, a coin cell battery holder and switch, and a Microchip ATtiny85 SoC. It's a good introduction to surface-mount soldering.

Once programmed via an Arduino, the chip's behaviour can be altered on the fly by connecting a wire between some of its pads. You can have a certain number of the LEDs illuminated to indicate the chip's value, or make them blink in a pattern.

Project

RIXE MARSEILLE

hsmag.cc/RixeGame

Ever fancied designing your own board game? Why not give it a go? It might not turn out to be the next Ticket to Ride, but you should have a lot of fun making and then playtesting it.

Maker Matlek opted to create a DIY Risk-like, territory-based board game based on the districts of their home city, Marseille. The map was designed in Inkscape, printed out, and then attached to pieces of cardboard mounted on self-adhesive black vinyl.

Game cards were printed on A4 linen paper, sprayed with lacquer, the fronts and backs folded together, then cut out – a technique shown on the very helpful Dining Table Print & Play YouTube channel: **hsmag.cc/BoardGameCards**. Playing pieces were 3D-printed, along with the game dice and a holder for them and the cards.



Project

ELECTRONIC DICE

hsmag.cc/ElectronicDice

Reckon those game dice are loaded, or can't be bothered to keep rolling them? This electronic dice system could provide the answer. Along with the ability to roll up to five standard dice, it features several built-in games: Poker, Craps, Yahtzee, and Duel.

Basing the project on the original Dicer by mbedni (**hsmag.cc/Dicer**), maker John Bradnam designed a PCB that houses the LEDs on the front, with IC sockets on the rear for the ATmega328 and MAX7219 chips. The former can be programmed via an Arduino Uno before plugging it into the board.

EAGLE files for the PCB, an STL for the 3D-printed case, and the Arduino code can be found via the link above.



MODELS & ROBOTS

Miniature and moving projects to build and enjoy playing with

Project

INTERNET OF LEGO CITY

internetoflego.com



Not only did Cory Guynn build a city out of LEGO, but he's linked it up to the internet. Built up bit by bit over time, it includes smart lighting, road crossings, motion sensors, and a railway.

A Raspberry Pi communicates with an array of Arduinos and other microcontroller boards operating numerous electronic parts. The IoT aspect includes fetching data from a weather API for a display in the city, along with a transport API for train scheduling.

The various parts of the LEGO city are covered in detail on Cory's blog, while code examples are available from his GitHub repo.

Project

FRED-209 NERF TANK

hsmag.cc/FRED209

Building a wheeled robot is a popular project. But how about adding some extra features? Having found a way to trigger Nerf foam dart guns with a servo motor, David Pride mounted one on top of a robot. Controlled using a wireless PS3 joypad, FRED-209 can fire multiple darts at a target.



The firing mechanism uses an original Nerf six-dart magazine with 3D-printed parts to launch the darts. A roller underneath can move along to tilt the whole mechanism up and down to aim.

Project

AUTOMATED MODEL RAILROAD

hsmag.cc/PicoRailroad

Microcontrollers offer an great way to control model railways. Kushagra Keshari is using a Raspberry Pi Pico to automate trains with his 'sensored' track (**hsmag.cc/SensoredTrack**). It uses an IR proximity sensor to detect a train passing and then, via a motor driver, gets it to speed up, slow down, or stop.



You could also connect relays to switch track points and reroute trains on a more elaborate layout and even set up automated scheduling.

Project

PICO DRAG RACE TRACK

hsmag.cc/issue51

Seeking a Raspberry Pi Pico project to amuse and enthuse his children, Robert Hamilton-MacLaren opted to use it as a race referee for a Hot Wheels model car track.

The build process is detailed in HackSpace issue 51, including using IR break-beam sensors on the finish line of each track lane to determine the race winner. It also features a DUPLO race tower with a set of red and green starting lights.



SUBSCRIBE TODAY

FOR JUST £10



Get three issues plus a
FREE Raspberry Pi Pico W
delivered to your door
(UK only)

hsmag.cc/FreePico

Subscription will continue quarterly unless cancelled

NEW INTERNATIONAL OFFERS!



Get 6 issues plus a **FREE Raspberry Pi Pico W** for just...

£35 USA and EU

£40 Rest of the world

Head to **hsmag.cc/subscribe** to get yours today!



FEATURE

HOW I MADE

GREENING THE SPARK

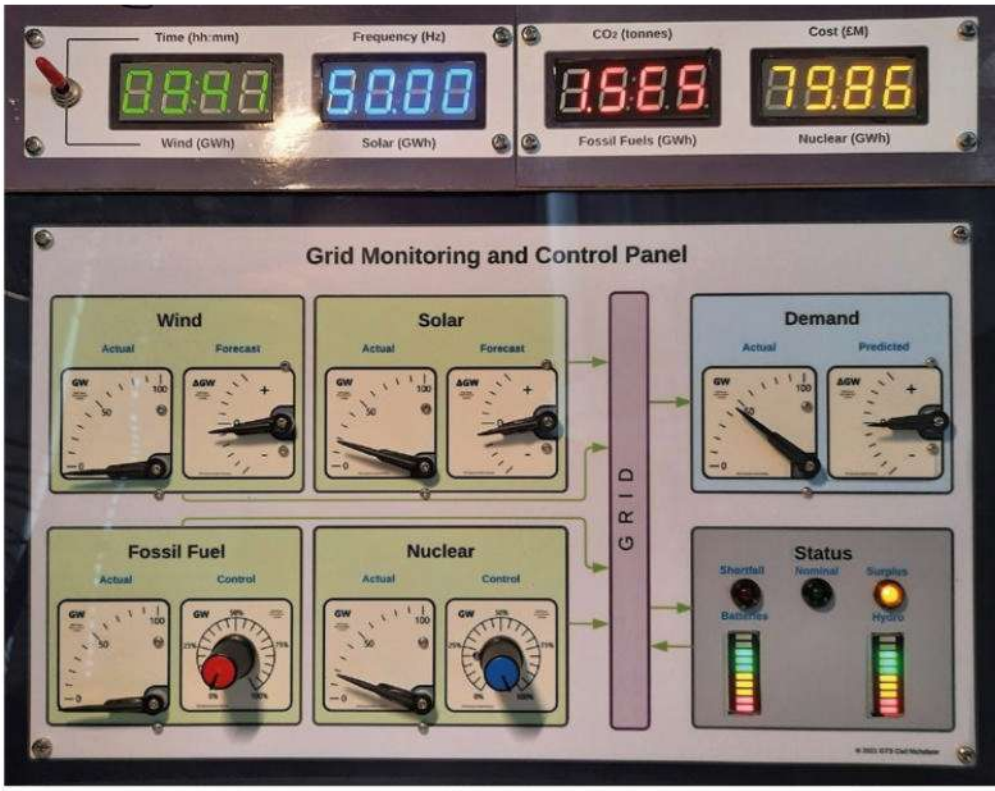
By Carl Nicholson

Greening the Spark is a model of a system for electricity generation, distribution, storage, and consumption – in short, a typical national grid. I have developed this as a private project over the last two years, starting with just a blank sheet of paper, a soldering iron, and some half-baked ideas. The current prototype is designed as a hands-on museum exhibit which can also be played as a game and is intended to educate and entertain on the subject of green electricity.

OVERVIEW

Hardware. Wind and solar farms are represented as a group of working model wind turbines and a sunlamp against a background picture of a wind and solar farm. Fossil fuel and nuclear power stations, batteries, and pumped hydro are themed static models whose functionality is modelled in software.

The operator interface to the grid monitoring and control system is a physical control panel with various displays and control inputs. There is also a visitor and staff control panel for setting up, starting and stopping operation. Everything is connected to a Raspberry Pi 4 Model B, running approximately 10,000 lines of Python.



Software models. Scenarios are presented as predefined timelines of weather (wind and sun) and consumer demand, with forecasts for each. The wind and solar farm models are driven by these weather models. Fossil fuels and nuclear power generation can be controlled manually by an operator or autonomously. Storage elements are controlled by an autonomous storage management system.

Operation. There are two modes of operation:

- Exhibit mode. This runs a simulation in real time and is a model of a national grid control centre with hands-on or autonomous control of fossil fuels and nuclear power generation. Manual control is initiated by touching one of the controls; autonomous control is resumed after a period of inactivity.
- Game mode. This involves controlling the grid over a ten-minute period played out at 144 times real time, so covering a whole day. The object is to adjust fossil fuels and nuclear power generated to avoid any blackouts or surplus power, to minimise either cost or carbon footprint, and to keep the storage levels at ~50%. There are three levels of difficulty (easy, moderate, and hard), and the results are displayed on the visitor control panel in the form of three 'e-sparks', which can be red, blue, or green, depending on overall performance. There is also a full simulation report which can be accessed via an online database, as described later.

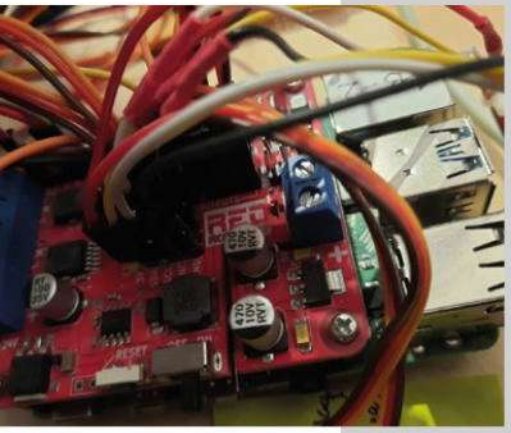
" THIS RUNS A SIMULATION IN REAL TIME "



Figure 1 Control your own power grid

Right Small-scale mock-up of possible landscape layout

The simulation runs in simulation time but the actual software runs in real time. The rate at which simulation time runs faster than real time is called the time factor, like fast forward on a video player. For example, when in game mode, the time factor is 144. Due to hardware limitations (the time taken for the ADCs to read the controls), the fastest loop iteration speed is 6Hz (real time) which is plenty for this project, where pinpoint precision isn't required.



Above Raspberry Pi with RedBoard+ HAT

Below Grid monitoring and control system information flow

CONTROL SYSTEM AND MODELS

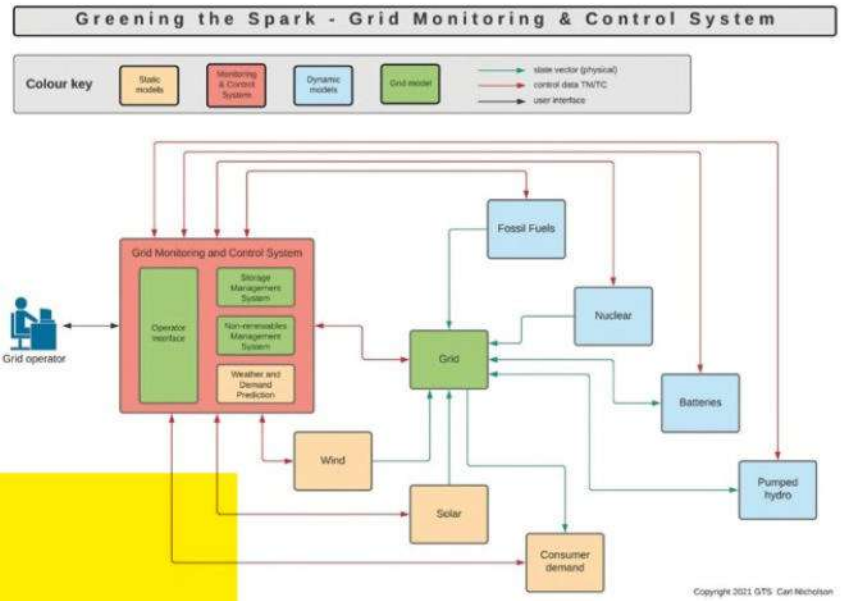
Grid controller operation is based on a spacecraft monitoring and control system (MCS) model which reads telemetry (TM) from the various subsystems, takes input from the grid controller, interprets the control laws, and then sends telecommands (TCs) back to those subsystems. This process repeats in a continuous loop until the simulation is finished.

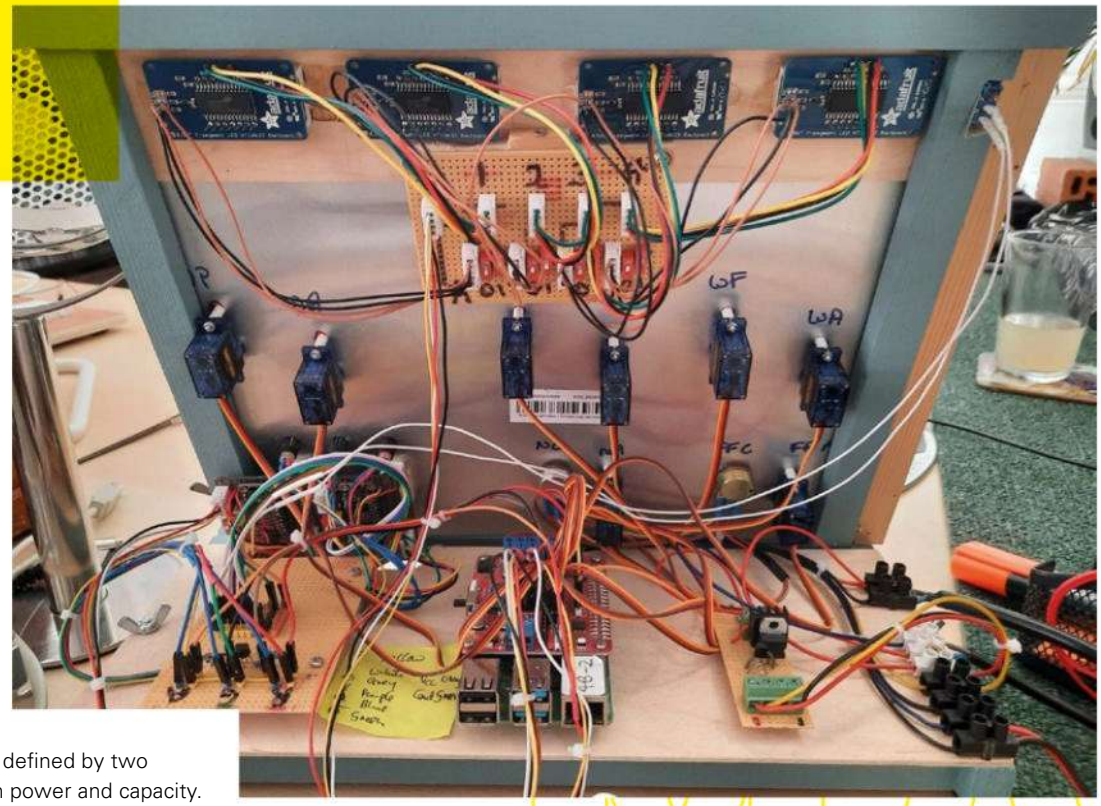
The MCS also contains a storage management system (SMS) which determines how energy is to be stored and retrieved in order to maintain a balanced grid. There is also a non-renewables management system (NRMS) for autonomous control of fossil fuels and nuclear. In autonomous mode, these sources are controlled without operator input. There is a semi-autonomous mode where the NRMS is in control by default and manual control is triggered by touching one of the controls. Autonomous control resumes when there has been no user input for a predetermined amount of time.

The weather and demand models are defined by uniform sets of samples over a timeline. The value at any given moment is linearly interpolated between its enclosing samples. The samples in a timeline file represent a baseline from which actual and forecast timelines are calculated. These derived timelines are calculated from the baseline, each run by addition of random Fourier components at DC, half-wave, full-wave, and sample frequencies whose coefficients are a function of the difficulty level – a game-type parameter affecting volatility and predictability of timelines as well as the scale of storage elements, the end result being to make controlling the grid less or more challenging.

> ***" THE FASTEST LOOP ITERATION SPEED IS 6HZ "***

Autonomous control (by the NMS) of the dynamic models can follow various strategies, depending on priorities and sophistication. A simple model follows the difference between renewables and demand and allocates power proportional to generators' capability. A better model takes storage into account. Other strategies include minimising cost or carbon dioxide emissions, or a compromise between the two.





Right →
There's a lot of wires to keep track of

Storage models are defined by two parameters: maximum power and capacity. Power is the rate at which a storage device can charge or discharge, and capacity is the total energy a device can store. The amount of energy stored as a fraction of the capacity is called the level, so a half-full battery has a level of 0.5, or 50% (the same as a half-empty one). There is always a discrepancy between generation and demand. The SMS manages the storage devices with a view to meeting demand (no shortfall) and avoiding waste (no surplus).

REPORTING – THE SIMULATION REPORT

Sparks are awarded on overall performance in three 'E' categories: efficiency, eco-friendliness, and economy, where a red spark is the worst, green is the best, and a blue spark is in between. Sparks are calculated as a weighted average of relevant points and converted to a colour. A report is generated for each game played, accessible in an online database by entering a unique report number (URN). A CSV file is also created for further processing/archiving. These reports contain the following summary data:

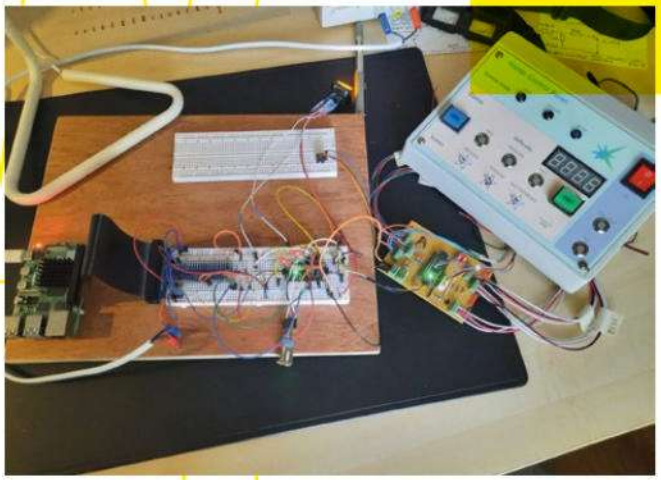
- Date and time
- Total energy generated, used, and stored
- Utilisation for each electricity source
- Total and average of costs incurred and CO_2 produced

Comments and scores are attached to quantities which indicate how well a grid operator has performed; they are generally calculated by comparing to a best or worst case and assigning points accordingly.

Timelines are included in the simulation report as three sets of data plotted against time over a 24-hour period:

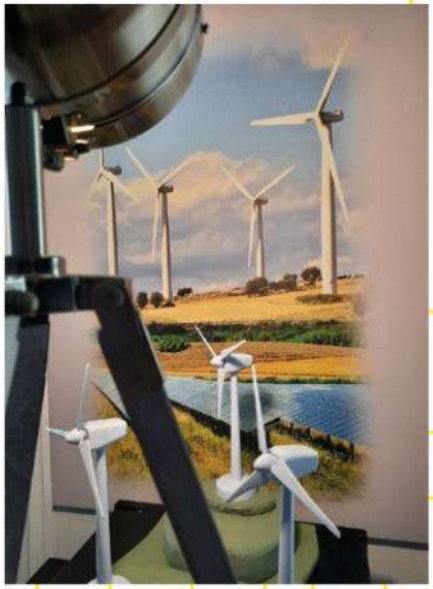
- **Inputs.** These are the given scenarios: wind-speed, solar intensity, and consumer demand
- **Controls.** These are the two generators the visitor has control over via the control panel: fossil fuels and nuclear
- **Outputs.** These are the consequences of the visitors' actions: how the power generated stacks up against demand, and the resulting state of the storage devices →

FEATURE



Right Visitor and staff control panel

Figure 2 Model wind turbines and sunlamp



AUDIO

Adding audio has given a whole new dimension to GTS in terms of ambience and narrative, bringing the installation alive. Sound effects and music are played as a background soundscape, an indication of something happening or a warning. Three characters inform (and entertain) staff and visitors with synthesised speech, courtesy of Google Cloud services. Music and effects are created with Ableton Live, with the final mastering being done in GoldWave.

HARDWARE

Power comes from a 12 V 20 A switch mode PSU with a custom 'glitch-buster' ultracapacitor (4*10 F@3 V in series) storage device, which can supply power for up to a minute in the event of power failure.

The Raspberry Pi is a model 4B with 8GB RAM and 32GB flash cards, fitted with a Red Robotics RedBoard+ general-purpose interface board (GPIO, I2C, ADCs, stepper motors, PWM controls). Access is provided by VNC Connect remotely from my desktop PC or laptop.

Operator control panel. The artwork was created in Adobe Illustrator and Lucidchart and printed on aluminium. This was machined to fit the components and mounted on a wooden frame. A numeric header was added later, using a wooden panel with artwork printed on glossy photo paper and glued to the panel, for four displays indicating either time, mains frequency, CO_2, and cost or power generated by each source. A switch to control which set of numbers is displayed on the numeric header. The various functions can be seen in **Figure 1**. Hardware types and connections are as follows:

- Four numeric displays are Adafruit four-digit, seven-segment displays with I2C backpack
- Numeric header switch is connected to a spare input on the batteries expander board
- Eight gauges are stepper motors connected to the stepper motor ports
- Two potentiometers (the control knobs) are connected to the ADC ports
- Three grid status LEDs are connected via a discrete NOR logic board requiring only two GPIO pins
- Two storage status ten-bar LED arrays are connected to an Adafruit GPIO expander on the I2C bus

" A NUMERIC HEADER WAS ADDED LATER "

Visitor (and staff) control panel. The artwork was created in Lucidchart, printed on glossy photo paper, and glued to an off-the-shelf equipment box. This was then machined to fit the components:

- One Adafruit four-digit, seven-segment displays with I2C backpack

- Two 12 V illuminated push-buttons (IPBs) connected via an I2C 16-channel GPIO expander
- Five 5 V IPBs connected via an I2C 16-channel GPIO expander
- Three monochrome LEDs connected via an I2C 16-channel GPIO expander
- Three RGB LEDs connected via an I2C 16-channel GPIO expander
- Mains on/off switch

Representations. There are representations of a wind farm with three model wind turbines and a sunlamp in front of a mural depicting actual wind and solar farms (**Figure 2**). These are controlled via the RedBoard+ PWM motor ports, the turbines directly, and the sunlamp via a custom Darlington Pair driver circuit (just visible in the bottom right of **Figure 3**) for extra oomph, getting its power directly from the PSU.

PROBLEMS

There have been many problems on this project (most of them software) – I couldn't possibly list them all here! These are two that caused me considerable pain and loss of sleep at a time when I thought they were going to be 'deal breakers'.

The Grove LED bar with backpack (Seeed Studio) would have been perfect, but I wanted to use it without the Grove Base HAT. As I was a beginner then, I couldn't work out how to do it, and didn't find its documentation or online support much help. So I abandoned it and built my own, based on an Adafruit AW9523 I2C GPIO expander. I think I could do it now, though, with its existing library and an I2C multiplexer.

Finding an MP3 player that worked proved to be problematical as various Python players (playsound, pygame) simply refused to work on the target Raspberry Pi. Finally I had to use OMXPlayer, which is called from Python using the standard OS system command method. Playing audio files couldn't be allowed to interfere with the operation of the simulator which runs in a continuous loop, so Python's capability for running parallel processes was used, also not without problems. In test programs, audio files would continue playing after the program terminated, resulting in the Python IDE crashing. Stopping audio at will has yet to be accomplished, because stopping the Python thread doesn't stop the operating system process!

WHAT (EVER) NEXT?

The website **greening-the-spark.com** is still a work in progress and should be online soon. Apart from that, there are many ways in which Greening the Spark can be developed, such as extending the range and sophistication of the physical and software models, better theming and artwork, web interface, live streaming of National Grid data, networking with other installations, advanced GUI design, AI-based control, data analysis and presentation – the list is endless.



Figure 3 Providing some simulated solar power

HackSpace magazine meets...

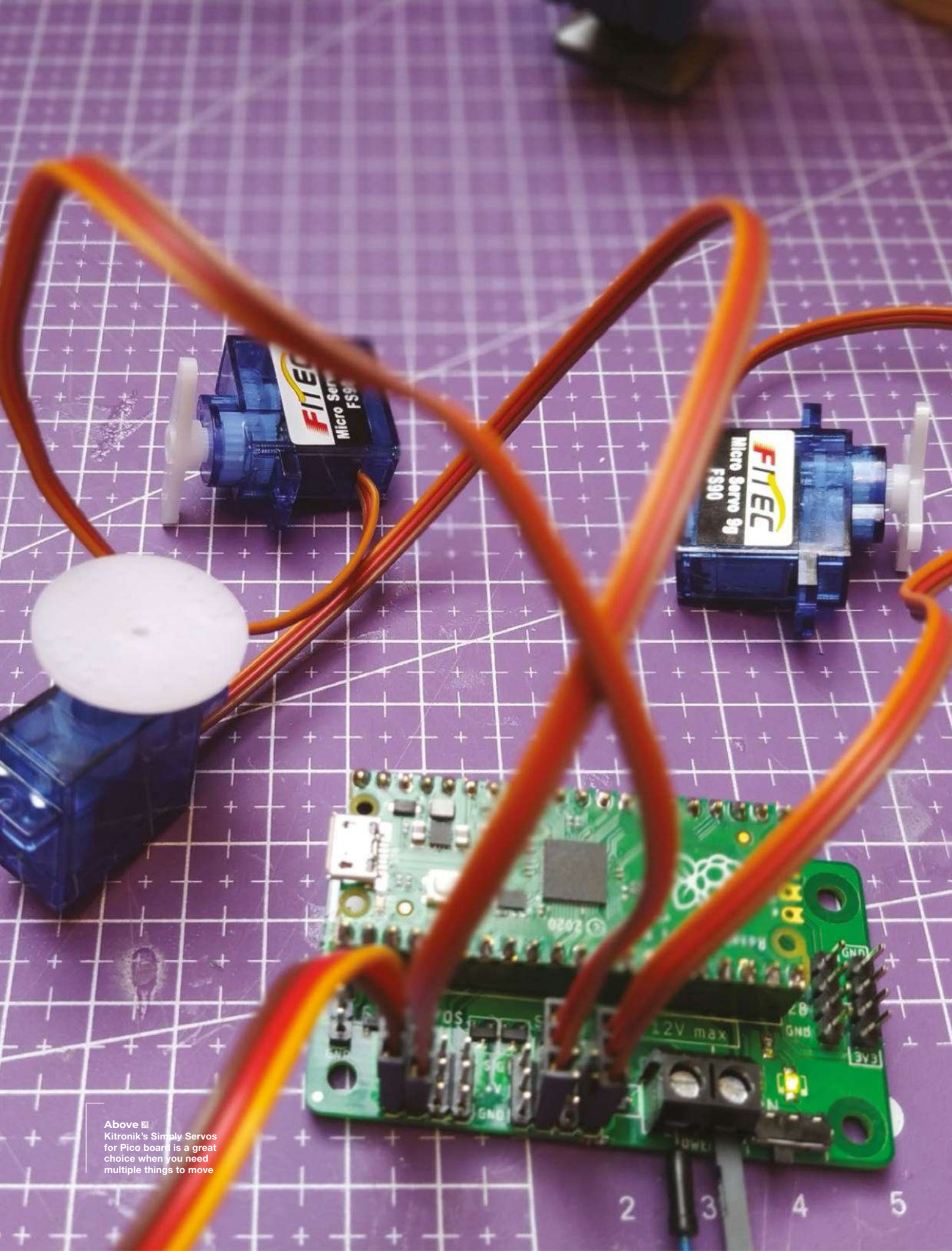
Kitronik

Kevin Spurr on getting electronics into schools

Kitronik exists to take the headaches away from computer science teachers. If they're scratching their heads wondering how to teach the computing curriculum, and they want a physical way to show what you can do with a few lines of Python (rather than the oh-so-boring 'Hello World'), teachers can get whole lessons' worth of gear to reuse time and again.

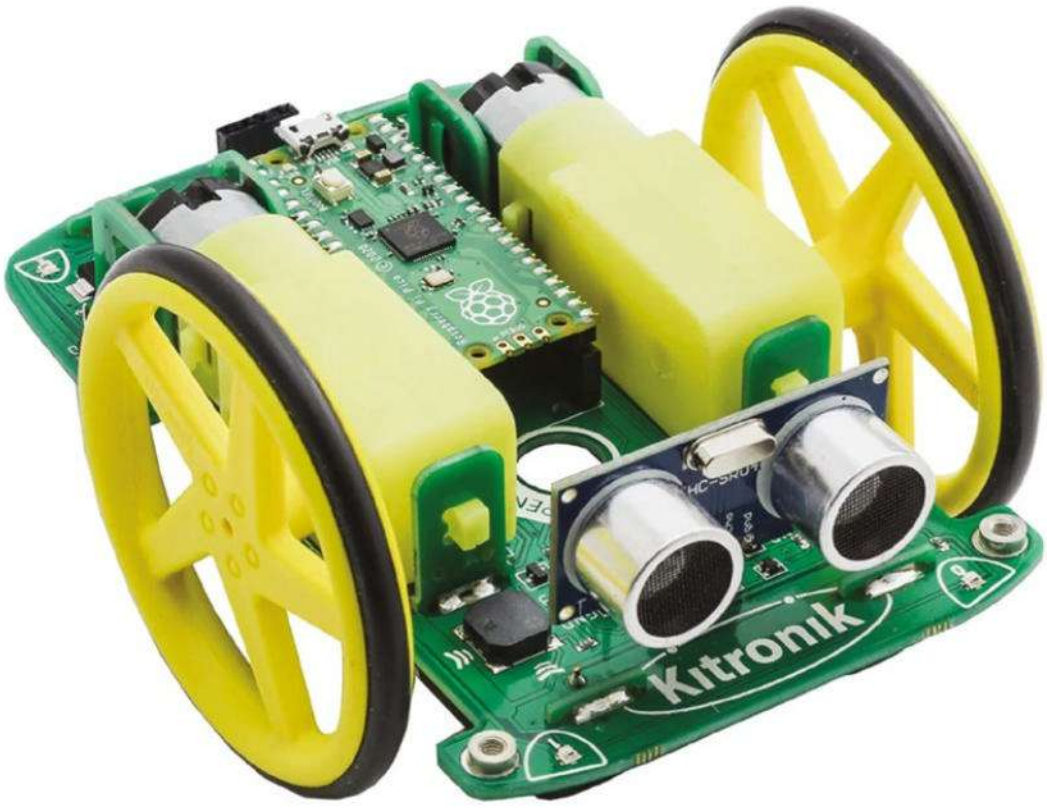
Its stuff isn't just for kids either: you may have spotted a few Kitronik products in the pages of HackSpace now and again, proving that you're allowed to play with smart greenhouses, buggies, motors, and the like even if you're a grown-up.

We spoke to Kitronik's co-founder Kevin Spurr to find out what they're about, why making electronics in Nottingham beats having a factory in China, and how the Pico and the micro:bit can fit side by side in schools.

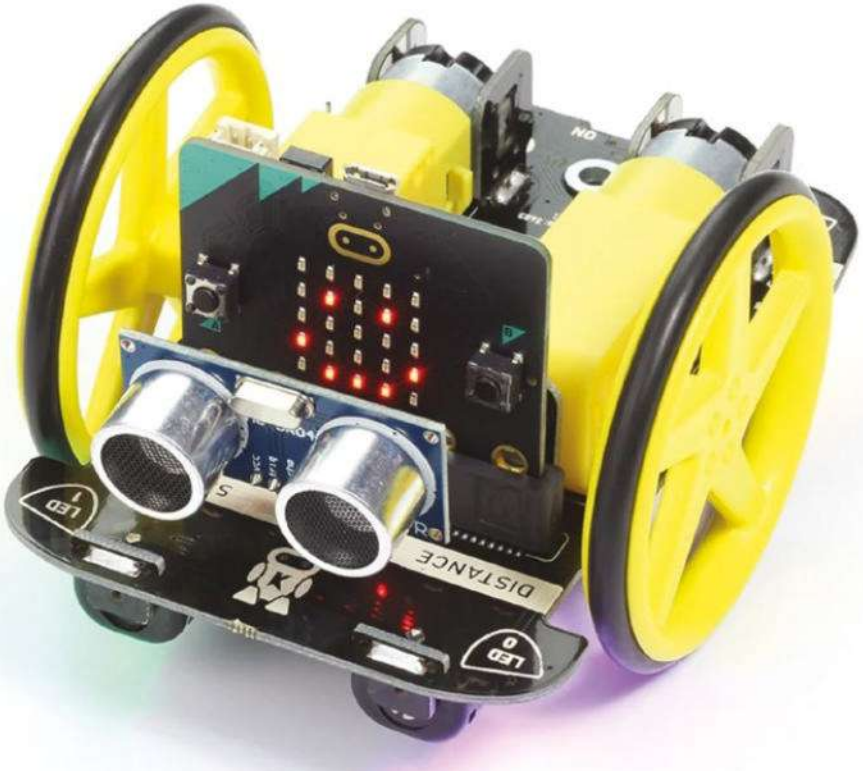


Above
Kitronik's Simply Servos for Pico board is a great choice when you need multiple things to move

INTERVIEW



Left
The Autonomous Robotics Platform (aka buggy) now comes in Pico as well as the original micro:bit flavours



HS Take us back to the beginning: when did you start, and what do you do?

Kevin Spurr Myself and Geoff Hampson started Kitronik back in 2005. We were working together at the time, with a background in electronics, design, embedded software, those kinds of things. At the time, Geoff was involved in a scout camp that ran every year. As part of that, he'd make a little project that he would take along, and the scouts would make, generally a solder-together project – there was no coding involved at that point. It might have been a lamp, or something else related to camping.

A lot of the other scout leaders happened to be teachers, so you'd get feedback on the projects and requests for things that they could do with their design and technology students. So, we did this for a number of years. After a while, we had a collection of these different projects. We looked around at what was being used in design and technology in schools and, actually, it was quite similar.

We realised that we could design some kits to sell to schools to make a bit of extra money. As it transpires, though, teachers and students need support. You can't just fire the product out and forget about it – you have people on the phone asking you questions, or asking for advice and things like that. So it quite quickly became a full-time job. We started off in a small unit, and we started designing more projects for design and technology. That's where Kitronik came from.

Over a number of years, we've extended that range to about 50 kits, covering all different kinds of subjects. All those kits have our teaching materials and, even today, that's still a big part of our business, supplying design and technology departments within schools for pupils aged eleven onwards.

HS Am I right in thinking that you're still based in Nottingham?

KS Yes we are. We're in a roughly 18,000-square-foot unit; we design the products here, we do all the procurement here, we do all the marketing, and we've got our own electronics manufacturing line here. The equipment we've got is as good as you'd get anywhere else in the world – we've got a Yamaha pick-and-place machine, screen printer, reflow ovens, all that kind of thing. A real production line.

I think when you get to a certain size, it gives you more control; you're not losing any margin out to a contract manufacturer. Plus, we can be more responsive: we don't have to buy really high quantities of any single thing. If we've got, I don't know, 300 units of a board on the shelf and someone places an order for 350, the next day, we could build that extra 50 and we can get that order out the door that week. It allows us to be more responsive, and also to keep our stockholding at a sensible level.

> "It allows us to be more responsive, and also to keep our stockholding at a sensible level"

HS How did you get involved in the micro:bit?

KS We were getting involved in some of the maker events like the UK Maker Faire. And, off the back of that, we got to know some of the people from the BBC. That's how we got involved in the micro:bit. We were one of the original companies that were involved in the micro:bit rollout. The micro:bit was designed to coincide with the change in the school curriculum, when it became less about using email or word processors to become more about learning to code.

The BBC had this big push on producing this device called the micro:bit to encourage that, because it's a physical device aimed at making that subject more fun, more hands-on. There were some big players involved: Arm; some big banks; and people like Microsoft. We came on board because we had experience of dealing with teachers and students. So these big companies, they do what they do, but that's not the expertise they had.

Every year, we sell to about 3000–3500 secondary schools in the UK, so we've got quite a good base of customers, and also people we can tap into. We helped out a lot there: we did some testing, and then we designed some resources. Because the micro:bit has that edge connector on the bottom, we then started thinking about motor drivers, so people could design buggies. Then we started thinking about other things, like an environmental sensor, or some other capability that the micro:bit doesn't have built-in, which we could add to it.

So we started to create these accessories for the micro:bit. And that also opened us up to a new customer base within schools, which was the computer science teachers, as opposed to just design and technology.

What we also found is that coding is more of a globally taught subject. Design and technology is actually quite a UK-centric subject – there are a few other countries that do it in a similar way like Australia does, and France used to – but coding is taught universally.

We now sell globally across many countries: all of Western Europe, Australia, North America, Asia – pretty much most parts of the world will have resellers selling our products.

But our focus is on education. Obviously, a lot of your readers will be makers, and I think some products are →

INTERVIEW

clearly designed for makers first, and they sometimes end up being used in education. That's great, but we are the reverse of that: we're trying to design for what a school would want to use first with a student or a teacher. And then, if it ends up in the hands of home hobbyists, that's great, but it's not our focus.

HS That's the reverse of a lot of the companies we deal with.

KS Yes, but then our products will never be as complicated. We're more focused on teaching resources, learning materials… that's where we put a lot of our effort.

We have a design team of four people here, and we also have two more people, so do a lot of work around videos, examples, tutorial builds for design, technology, or for coding. We put a lot of effort into that side of things. We don't say: 'here's a board, here's an example program, off you go and create your own'; that wouldn't really work in an education setting. There needs to be more of a focus on the resources and how they progress from not knowing very much through to being able to create something more complex in a step-by-step fashion.

HS Do you find that you get a lot of orders from hobbyists?

KS They do; they tend to order, say, a motor driver board or a particular type of sensor, or something along those lines, rather than the classroom solutions.

I wouldn't say it's our predominant market; we're probably 90% education-focused, and the other 10% is people online buying things in small numbers.

But we don't try and compete with people like Pimoroni. I know Jon and Paul there really well. They will hold vastly more product lines than we ever want to. But that works for their customers, doesn't it?



Below
Grow herbs in a micro:bit-powered smart greenhouse

We're trying to focus on [8- to 14-year-olds, that's] probably our core market. If you're learning coding beyond 14, you're probably going to step away from coding a buggy to drive around the floor; you'll probably move on to something like a database design, websites, or something that's more focused on just using a computer.

HS Which is a shame, though, because it's vastly more fun making something that moves. Maybe they'll get back into robotics when they have a bit more spare time or money on their hands.

KS That's one thing we are very aware of: we always try to make our products accessible price-wise. We're not trying to create some little board that you need a really good job to buy.

The Raspberry Pi Pico is a great example: we've got a little discovery kit for that which is £12. You can learn to code it, you build it on a breadboard. There's a tutorial book in there, and afterwards you've still got the components, so can go and create your own things. It's affordable for anybody.

HS You've been in the business of supplying teachers and classrooms for a while now. Have you seen the types of things that they build change over that time? Kids today must bring a lot more to the table now than they did ten years ago.

KS Things like the use of Scratch and the MakeCode environment have made coding far more accessible, particularly in junior schools. So they're coming to secondary schools with a better knowledge base, and they'll probably carry on with some block coding for a while, but then move on to things like Python, which is one we're seeing a lot of. Micro:bit also does a very good drag-and-drop Python editor.

We use the Thonny editor for our Pico products. But the micro:bit drag-and-drop editor is a great step in between, because – like blocks – it allows you to take a prewritten section of code, an iterative

loop or a for loop or something, take the bits you need, and you can drag them in and then put your own values into it.

So the format you'd need is already shown to you. It can be quite daunting to go from block coding straight into a textual language. That transition is something that takes some working through.

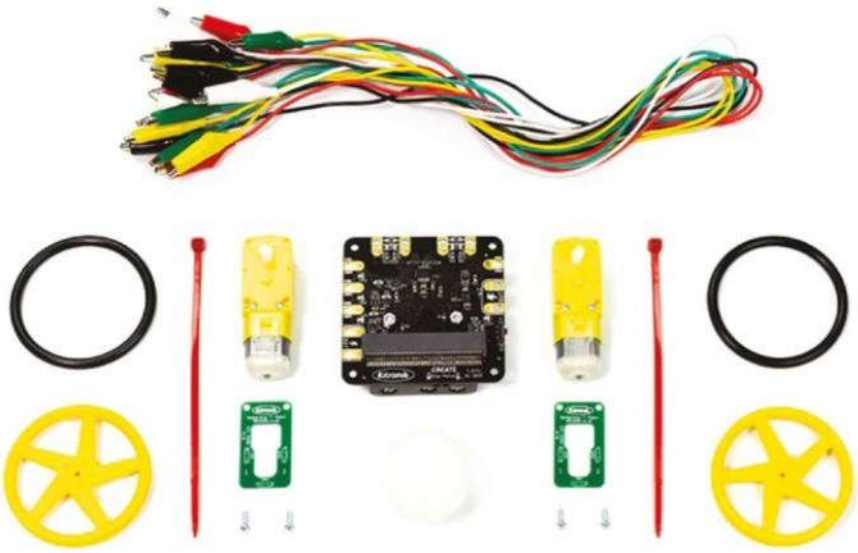
HS And I guess another good thing about that is that you're not wasting your time learning a kids' version of coding: it's all valid, usable stuff that can control hardware.

KS It's really, really good. And it simulates on there as well, so you don't even need a micro:bit – you could just run code on the simulator, if you wanted to get used to that.

Another thing we're seeing is that more complex technology is becoming much more easily accessible. Before the micro:bit came out, controlling some of these complicated boards and things could be quite a challenge. The barriers to entry were high. But over time, things are becoming much, much simpler. You can get boards for doing IoT quite readily at a low cost. Even the Raspberry Pi Pico – put that onto a Wi-Fi network, and you've got the guts of an IoT build. You couldn't do that for £4 ten years ago.

HS I guess another barrier to entry that's come down is that ten years ago you would have had to learn to program in C.

KS That's what I did. When I was coding, you'd have a £1 processor. Great, but then you'd have to go and buy a £100 dev kit for it, because you couldn't just plug it into USB and code it. So you'd have some dev kit that you'd have to buy, and then you'd possibly have to buy a coding environment, because there weren't any good ones for free. Just to get into doing a bit of software and hardware, you'd need to have a bit of money behind you. Whereas nowadays, you don't: just plug a USB lead in and off you go.

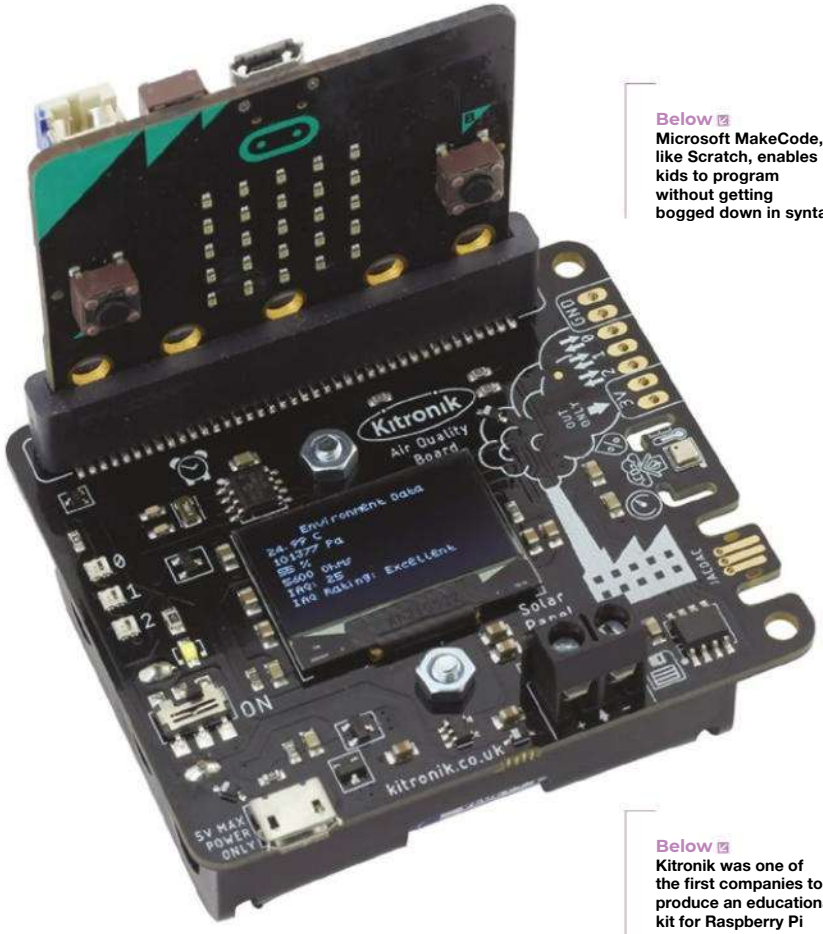


> "Another thing we're seeing is that more complex technology is becoming much more easily accessible"



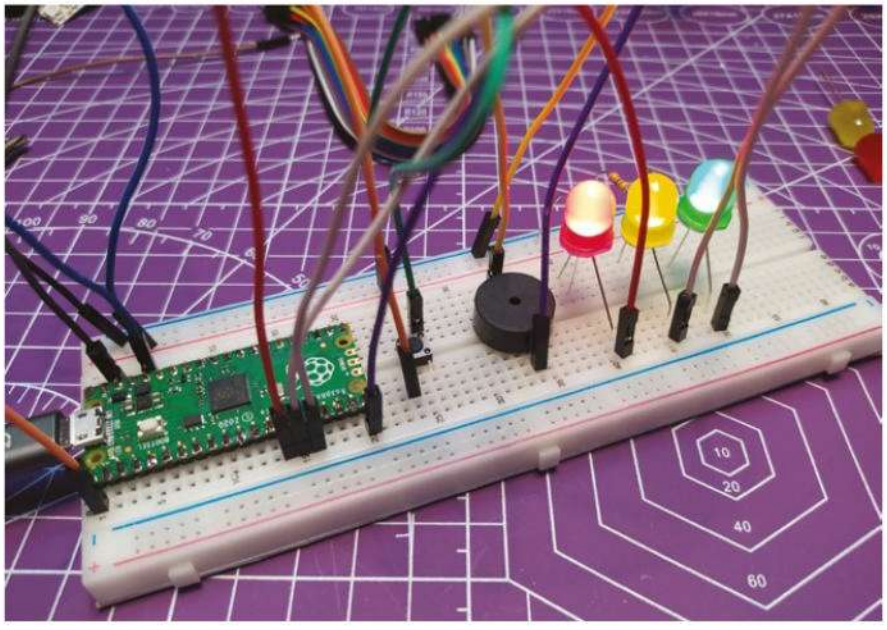
Above
One of Kitronik's aims is to let teachers concentrate on teaching. This is one of their lessons in a box, a full robotics kit with lesson plan for schools

INTERVIEW



Below
Microsoft MakeCode, like Scratch, enables kids to program without getting bogged down in syntax

Below
Kitronik was one of the first companies to produce an educational kit for Raspberry Pi Pico, seen here in its classic traffic-light configuration



HS What sort of feedback do you get from teachers?

KS The feedback is good, but they're looking for more of an overall term of work, a whole project that they can run. One of the barriers we come up against is computing teachers who maybe aren't that confident. Because it's a relatively new subject, there's not a massive depth of teaching knowledge there that's been around for decades.

ICT was traditionally taught as a second subject by a PE teacher or a business studies teacher, and yeah, they could teach you how to use email or something, but then suddenly teaching kids to code… that's a different ball game.

We've been doing a lot of work trying to train and upskill teachers, as have various other organisations, but still, we don't want to assume too much knowledge on the teachers' behalf. So, we want to try and deliver them a package of work that they can use to make a fun term of work with their students.

We can take some of the worry away, because they know that the hardware is going to work. These are lesson plans that have already been tried with teachers… we can provide them with a solution so that they can get on with the teaching element. That's what we're trying to focus on.

HS I know a head teacher who was absolutely beside herself in 2013 when the new curriculum came in. I think she just crossed her fingers.

KS That is a lot of what we're thinking about, which is why we're never going to put out hundreds of products. We're trying to hit curriculum themes, with all the resources and everything that will make it usable.

And then I think the knock-on effect of that is that, when an individual person buys it, they also find that they have a good time with the product. Because we've tried to put a bit more effort into that side than maybe other people would do.

HS If you're selling globally, does that mean that you need to keep up with school curricula in different countries?

KS That is a challenge! I was in North America recently for ten days. I went to visit a lot of my customers over there.

What you find is that the learning elements are repeated, but the theming might be slightly different in different countries. If you're in a country that's very big into farming, you might find there's a lot of agricultural themes within their STEM learning. Some places will focus more on engineering or robotics.

So, I guess what we're trying to do as a company is identify the big themes globally, and try to design for them. And then, hopefully, we'd have a set there that teachers can pick and choose from. Some areas are popular everywhere – things like AI, IoT; they're things we hear about a lot.

The challenge is to try and find a way of teaching in a fun way, but also a deliverable way. For example, IoT is great – there's demand for it from teachers. But schools won't let you put [IoT projects] on to school networks, because schools are very protective of their networks. There's a lot of data protection and safety around children.

They might let you put a board onto the network, but it might have to have an antivirus package on it – well, good luck putting that onto a Pico or even a [Raspberry] Pi at times. So that's the kind of challenge we're trying to address. IoT is a great thing to learn about, but how do we make it so that it can be delivered successfully in the classroom with all the IT challenges that there are in a school environment?

HS Are all schools receptive to your approach? Or are there still some holdouts saying that they are sticking with flashing an LED on and off?

KS There's no specific requirement in the curriculum to teach with hardware, so if a school has a suite of Windows PCs, they might decide to do it all on the screen, which I don't think is as engaging for a student. But you know, if a school doesn't have any money, but it does have a room full of computers, that's an asset that they can use.

You do find that there's a lot more coding with physical hardware in the junior schools. With younger children, schools probably don't have a room full of computers that you might see in a secondary school. With younger kids, they'll get tablets or laptops out, and they'll probably want to share that between a few kids. That physical computing is very much something that you see all the time in junior schools.

HS What sort of things do you make for Raspberry Pi Pico?

> "The challenge is to try and find a way of teaching in a fun way, but also a deliverable way"

KS We've done a few things that have traditionally worked well with the micro:bit; we did an inventor's kit where you build on a breadboard – we've converted that to a Pico version. We've got a really nice environmental board which is great for education; it does pressure, CO_2, temperature, and you can log that into the memory on the Pico and extract that information later.

It's got a real-time clock on it as well, so you can time-stamp the data. A student might say: does the air quality in my classroom vary during the day? Let's log that information, then I can put it into Excel and see what's happened to the air quality index, CO_2 levels, all those kinds of things. What happens when we open the windows during the day? Does that improve the air quality? What happens if we put the central heating on? If my bedroom faces onto a busy road with lots of cars, is my air quality worse than maybe another child who lives somewhere more rural? There are all kinds of things that they can do with that environmental board.

That version for the Pico as well, I'm pretty sure it's got a high-power output on it as well. So, if you wanted to have a little heater, or some kind of high-powered fan, you could control that from that board. Which makes sense when there's environmental control: you might put it in a greenhouse and use a servo to open the window when it gets too hot, or switch on a heater when it gets too cold.

With the Pico W, you can do a lot of that remotely now, dragging the information back across a wireless network. We've also converted our autonomous robotic buggy that we built for the micro:bit so that it now works with a Pico. We've got an example whereby we've written a touchscreen app for a computer where you can control the buggy over Wi-Fi.

More generally, we see the Pico being good for progression after learning micro:bit; that's why we like it.

On a micro:bit you can do a bit of Python programming, but at some point, you'll probably want to move on to something that can do larger Python programs, and we see the Pico as a very good step for that. It works well in schools because they're cheap, you can embed them into projects, and you can just code them with a normal computer which the teachers are used to using.

The Pico is a good little board. I guess a lot of people will probably go down the route of putting the chip used in the Pico on their own products – I know the likes of Pimoroni and Adafruit have been doing that. I'm not totally sure that's a way we'll ever go, because, in a school setting, the ability to just pull a Pico out of an old project and put it into a different project is really nice. □

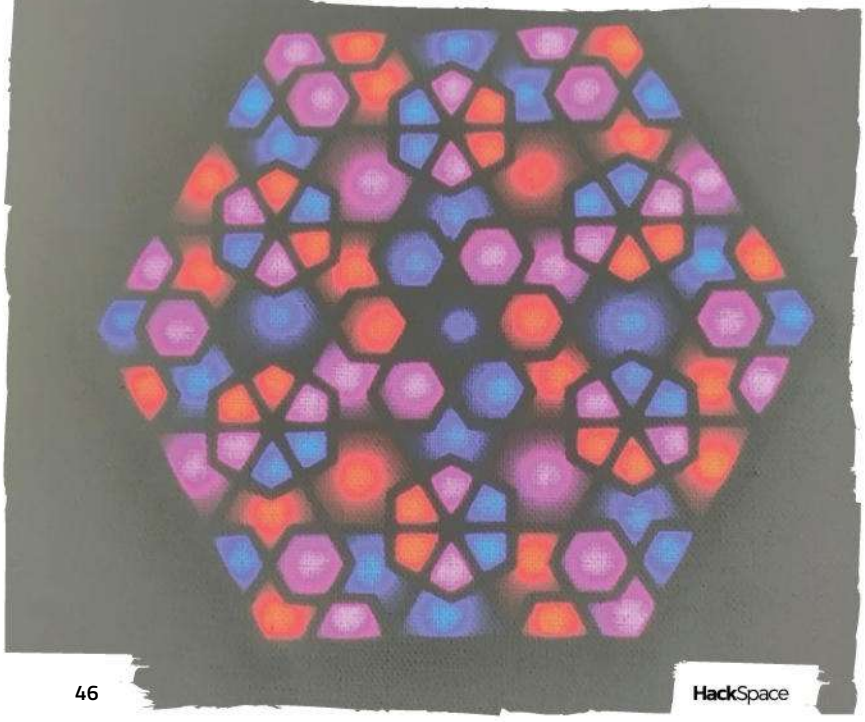
FEATURE

IN THE WORKSHOP: Geometric art

By Ben Everard

A Hackspace collaboration

Why are LEDs always in grids? There are, of course, much bigger things to get worked up about, but for some reason, this bothers me. There's nothing about a point of light that means that it has to light up a square, and there are far more interesting ways to capture that bit of light. This is a long-running bug of mine.



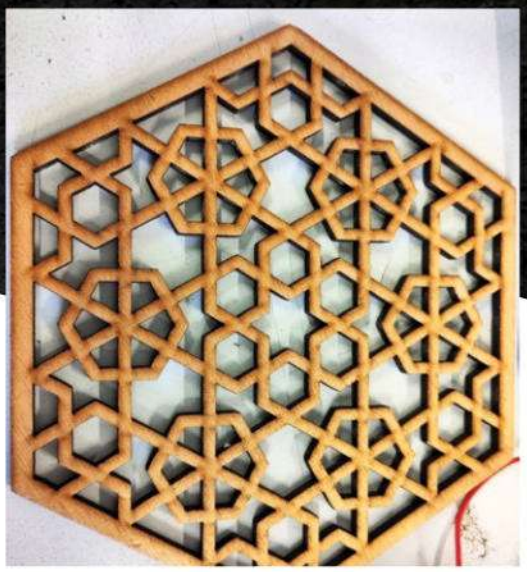
Below
The LEDs in larger sectionsseem dimmer. I should fix this in the software

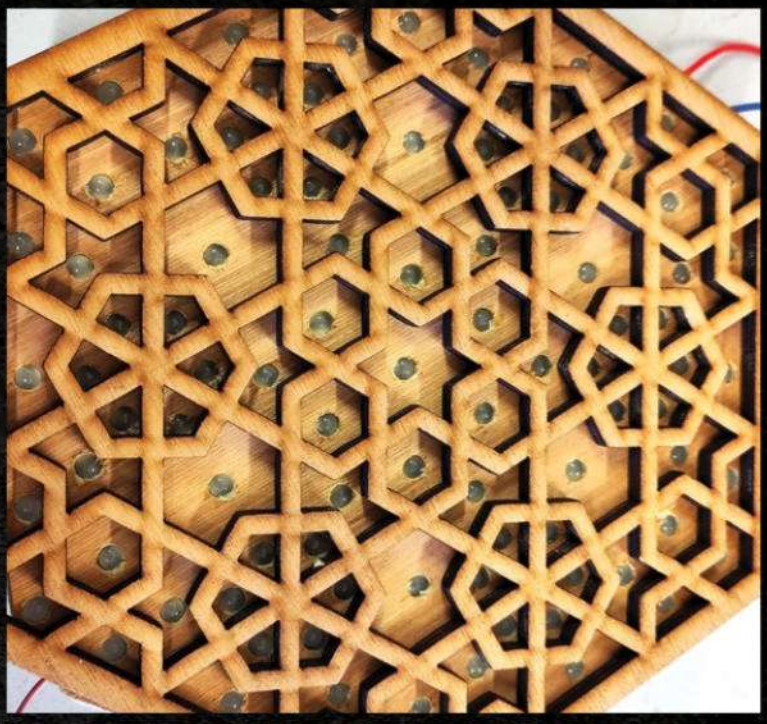
A bit more than a year ago, I was working in Bristol Hackspace trying (with some success) to get the old 1980s-era plotter working with Inkscape. Another member – Riaz Shafi – asked for help plotting one of his projects, an Islamic geometric pattern that he was working on. In the end, he found the laser cutter a better fit for his work, but we started chatting about art, light, and patterns. At the time, I was exploring artistic PCB design, and wanted to see if I could do something with this in the style of Jason Coon's LED disks based on the Fibonacci pattern.

Using one of Riaz's patterns, I created a concept PCB, but it felt like that approach wouldn't work. The basic problem was that the size of each 'pixel' varied significantly, so even if the smallest pixel was shrunk so that it only just fitted an LED, the overall size of the PCB became so large that it didn't make economic sense to go down this route.

The idea sat for a while. The problem was that we needed a way of building the thing so that it didn't end up taking forever to wire the thing up (I'm perhaps still a little burned by a 3D LED sculpture that I made a while ago that was so small that, despite getting all the LEDs into it, I never actually managed to wire it up – even with tweezers).

While browsing AliExpress looking for LEDs for another project, I came across a string of 5 mm through-hole WS2812B-compatible LEDs. I'd not seen these before (the 12 mm WS2811s are common ones that I've used a lot, but not these smaller ones). These would be easy to work with – they'd just need





a drilled, or laser-cut, hole and could then be pushed into place.

I ordered a set, and started work using the same design that I'd used before.

WS2812B LEDs are easily controllable from just about any microcontroller, and I used a Raspberry Pi Pico.

DIFFUSION AND SYMMETRY

Just putting LEDs in the right place doesn't give you the right effect. You also need something to diffuse the light. During testing, I just laid a sheet of A4 paper over the separation layer. Regular printer paper actually makes a really good LED diffuser, but it's not very strong and, even though this was originally a test piece, I ended up wanting something a little more permanent. Artists' canvases give a really good diffusion effect. There's a slight texture to the fabric that shines through and, when the LEDs are off, the blank canvas looks exciting.

There are libraries of hundreds of patterns for LEDs that you can use. From the ubiquitous rainbows to fire effects. However, despite testing many of these, they don't really look good in a geometric pattern. Certainly not this geometric pattern.

It quickly became apparent that, in order for an LED pattern to look good, it had to respect the geometry of the underlying art. One of the key features of which is the symmetry. Once I started to play with symmetric patterns, things started to look better.

I've played with a lot of patterns, but the one I'm running at the moment builds up a pattern of three colours placed symmetrically on the canvas. They build up in an ordered sort of way, that looks kind of chaotic, but are blanked out after a few seconds in a more ordered fashion that highlights the features within the pattern.

I wanted to make this piece interactive. My first idea was to let people have some control over the colours, and I added three potentiometers which increased the chances of red, green, and blue appearing in the random colours, but overall, I don't think this adds much, so I removed it.

I'm also considering the option of adding touch input to it to allow you to change certain elements and see that change reflected through the symmetry of the pattern, but I'm not quite sure how to make that work. So, for now, it's happily cycling its way through random patterns.

This piece was only a test piece. Riaz took the techniques we tested here and built a larger piece with 400 LEDs displaying his own pattern. ◘



Above
A laser-cut grill separates the LED segments

Left
I placed the LEDs by manually drilling holes, but this could easily be laser-cut

Left
A chain of LEDs is easy to wire up

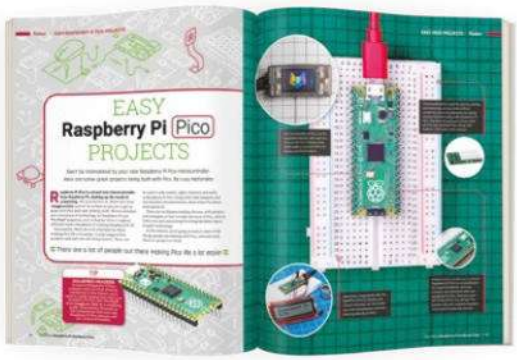
THE *Official* RASPBERRY PI HANDBOOK 2022

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects







Buy online: magpi.cc/store

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 52
SURFACE-MOUNT SOLDERING
Holes? Where we're going, we don't need holes

PG 58
TIDE TRACKING
The sea didn't wait for Cnut, and won't wait for you

PG 64
CAPACITOR PLANES
Electric planes without batteries



PG 72
METAL STAMPING
Tap-tap words into steel

PG 50
SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

50 Flexible 3D prints

PG 76
BLUETOOTH
Debug your projects remotely

PG 78
PRINT FAST
Slicing for speed



Flexible 3D printer filament

Bendy prints with TPU



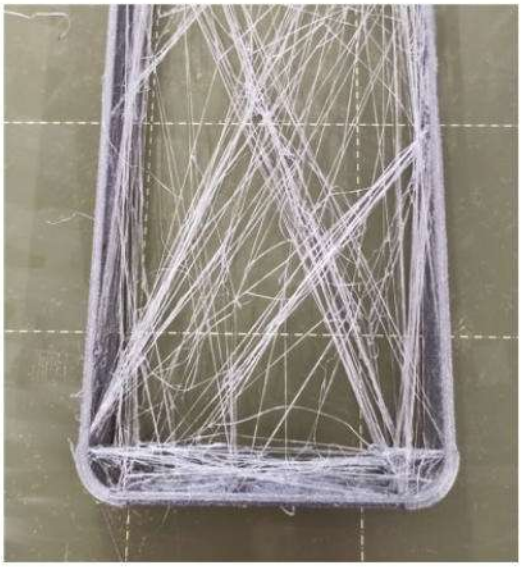
Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

The first thing you need to know about thermoplastic polyurethane (TPU) is how incredibly variable it is. With most other printer filament – say PLA or PETG – you get different brands and different qualities, but they're broadly similar and generally print the same way. TPU is different. It spans the range from that which is as wobbly as cooked spaghetti to filament that is about as stiff as PETG.

TPU is generally known for its flexibility – in fact, it's often called flex filament. The flexibility is given as a Shore Hardness rating, which can be confusing because there are two scales commonly used with 3D printer filament: A and D. These are different scales made in slightly different ways, so they're not directly comparable. Each scale runs from 0 to 100, with 100 being the hardest. Generally, D is used for harder materials because 100D is harder than 100A.



Right
TPU is typically quite prone to stringing, but this is an extreme case

Most flexible filaments are in a range between about 90A and 98A. However, the softest material we've got for our tests is Fiberlogy FIBERFLEX, which is 30D. This is noticeably softer than the 90A TPU. At the other end of the spectrum, there's NinjaTek Armadillo, which is rated 75D and is noticeably stiffer than the 98A-rated Flexfill from Fillamentum.

The actual flexibility of the filament is only half the story, though. How you print a piece will also have a big impact on its flexibility. For example, a part with less infill, but in a stiffer filament, might feel more flexible than the same part in softer plastic but with more infill.

So, that's the basics of the plastic – let's look at how you actually print it.

Once you can get TPU through the hotend, it generally prints quite well. It sticks to most build plates well (so well, in fact, that you might need to add glue stick or similar to help get it off when you've finished). However, the challenge is that it's so soft and bendy that it doesn't always play well with the extruder gears.

The big split between printers in terms of their performance with TPU is whether they have a direct drive (i.e. the motor that pushes the filament is located on the print head) or Bowden tube (the motor that drives the filament is located away from the print head and is connected by a Teflon tube). When working with TPU and a Bowden tube printer, physics is working against you. When the extruder motor drives filament, it has a tendency to bend and flex and squash, rather than going into the extruder. The extruder motor also retracts when the print head is moving; when this happens, the filament can just stretch. The longer the Bowden tube, the worse the problems.

This doesn't mean that you can't use TPU on a Bowden printer; it just means that you have to be careful. Working with harder TPUs will be easier (you can still print with fewer walls or less infill to make your print flexible).





Left
Prints with TPU can be incredibly flexible. This Polymaker PolyFlex 90A can be crushed up into a little ball, then it springs back to its normal shape

There aren't any hard and fast rules about this. If you want to find the limits of your printer, we'd recommend getting some filament samples. These often only cost a few pounds, and are a great way of finding out how well something prints without committing to a full roll.

The basic technique for printing TPU is the same regardless of its softness – print slowly. How slowly will depend on the model, the printer, and the filament. Start with the recommended settings for your printer and if it seems to be struggling, slow down.

When designing for TPU, bear in mind that the part of it that's already printed will be flexible. The more flexible the filament, and the narrower and taller the model, the more it's going to flex as additional filament goes down on top.

WHAT TO PRINT

So, now you've got some TPU and perfected your print parameters, what exactly should you print? Here are a few of our favourite TPU projects:

- Phone cases. Most of us carry expensive computers in our pockets, and many of us are prone to dropping them. You can buy cases, but a 3D-printed case can be personalised. You can pick the colour you want and put a design on it.
- Robot tyres. Different robots need different tread patterns and sizes of tyre. There aren't many options available off the shelf, so 3D-printing a TPU tyre is a great way of getting exactly what you need.
- Cable ties are widely available, but if you have particular needs that you just can't get elsewhere, you can 3D-print your own.
- Wallets. Let's be honest, 3D printing is not a good way of making a wallet, but if you really want to show off your nerd cred, the option is there.

Finally, TPU prints generally need quite a bit of cleaning up. They're likely to be covered in strings and clumps of filament. You might be able to minimise these with print settings, but you may not be able to eliminate them entirely. It's not a big problem, though, as you can shave these off with a sharp knife.

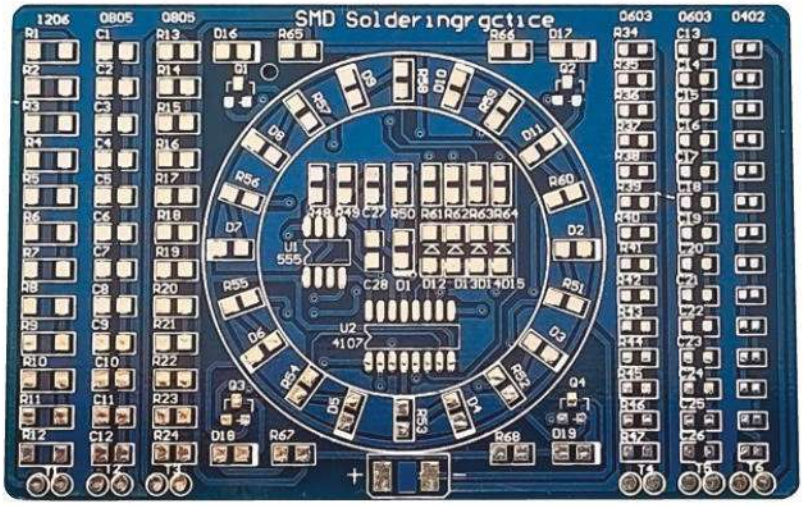
There's something exciting about printing in flexible material, maybe because there are very few other ways of making something flexible. There's something about flexible things that feels a bit more space age. The fact that you can vary the squishiness of the thing by varying the number of walls and infill percentage only adds to this. What will you print and squish?

Below
Small strings and blobs can be quickly and easily shaved off with a craft knife



Surface-mount soldering

Don't be afraid of this essential maker skill





Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Figure 1
This is an excellent choice for learning to surface-mount solder

Through-hole soldering has long been a staple of the maker community, but the rest of the world has moved on. Electronics products that you buy in stores have very few – if any – through-hole components on them, and it can be hard to find modern parts in through-hole packages. If you work with electronics, sooner or later you're going to need to solder something without legs.

Surface-mount soldering has a reputation for being hard, but it really needn't be. In fact, we'd argue that in some ways it's easier because nothing's hidden halfway through the board as it can be in through-hole soldering. Yes, the parts can be really small, but they don't have to be.

Let's start by first making sure we understand the parts we're working with.

There are a few different types of components, and we have to treat them differently. First, there are components with two pins – mostly resistors, capacitors, and diodes – which are typically rectangles with pads at each end.

These come in many different sizes and are usually denoted by four numbers specifying their width and length. Unfortunately, these numbers can be in either inches or millimetres – it's never specified which.

The standard sizes are:

Imperial	Metric
1206	3216
0805	2012
0603	1608
0402	1005
0201	0603

The largest of these is 0.12 × 0.06 inches (or 3.2 × 2.5 mm in sensible units), and get smaller from there. There are both larger and smaller sizes, but these are the most common ones used in hobbyist projects.

For components with more than two connections, there are a few different package types that they can come in.

Small outline transistors (SOTs) usually have three to eight pins split someway between two sides of the package. Each connection is a small pin bent flat, so the whole package sits flat on the surface.

Small outline integrated circuits (SOICs) look remarkably like dual in-line packages (i.e. through-hole chips) that have the pins bent flat for surface-mount soldering.

Quad flat packages (QFPs) are typically smaller than SOICs and have pins on each of the four sides rather than just two.

There are also a few types of surface-mount devices that aren't easy to hand-solder. These are quad flat no-leads (QFNs), the rarer dual flat no-leads



Left
The basic equipment for surface-mount soldering: iron, wick, solder, and tweezers

(DFNs), and ball grid arrays (BGAs). In each case, the pads are under the component itself. With QFN and DFN, the pads are only just under, so they're not necessarily impossible to hand-solder, but they're pretty tricky and we won't be looking at them here. You can work with these using a hot air gun, hot plate, or soldering oven, and we'll look at doing this in a future tutorial.

EQUIPMENT

Soldering iron. Contrary to popular belief, you don't need a particularly fine tip on your soldering iron. Conical tips are a good choice for getting started.

Tweezers. You need these to handle the small components. There are a few different types available, and it really comes down to personal preference. We prefer angled tweezers. Electrostatic discharge (ESD) tweezers are electrostatically safe, and it's probably best to get these if you can.

Solder. We'll be working with solder wire (we'll look at solder paste in a future tutorial). It's easier to use finer diameter wire (such as 0.5mm) if possible, but you can use thicker solder if that's what you have.

Desoldering wick. You will almost certainly end up bridging pins when soldering surface-mount components, so some way of removing solder is needed. We prefer wick, but you could use something else.

These are the key bits we'll be using. There are a few other bits you might want to consider:

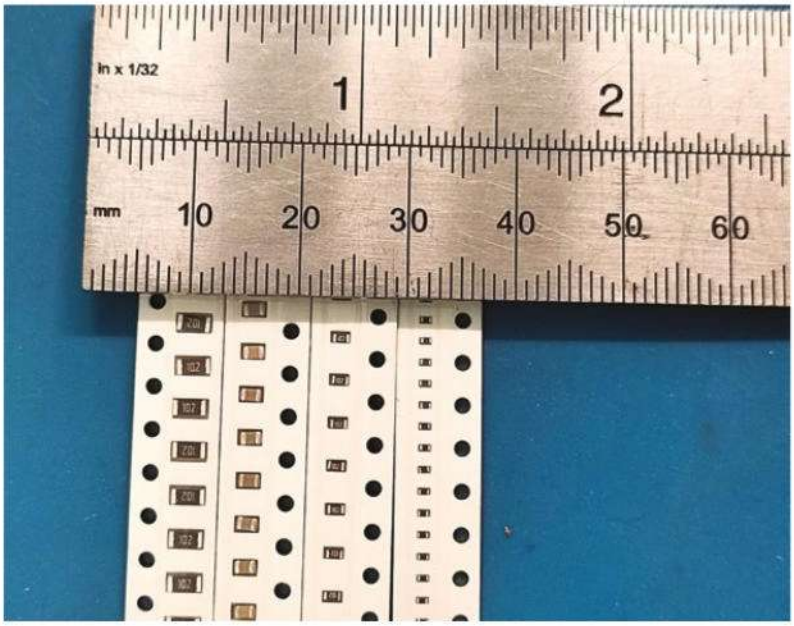
Fume extractor. As the parts get smaller, you'll be moving your face closer to the workpiece, so you'll be more prone to getting a face full of toxic smoke.

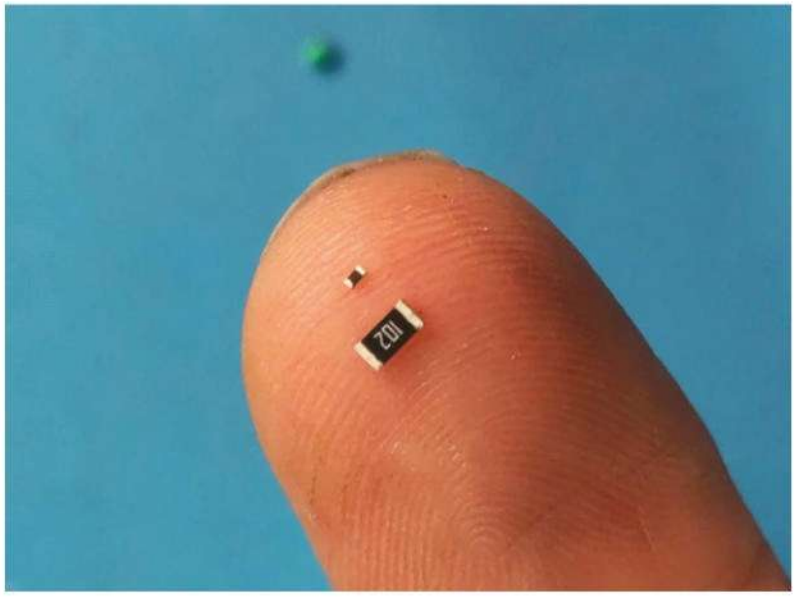
> **Contrary to popular belief, you don't need a** particularly fine tip on your soldering iron

Flux. It is possible to solder surface-mount components without flux – particularly the ones with two pads – but we wouldn't recommend it. Flux comes in a few forms. We prefer flux paste, but flux pens are also popular.

Magnifiers/microscopes. When you work with smaller parts, it can be hard to see what's going on. There are particular features you need, particularly with microscopes – stereoscopic scopes are much →

Below
1206 down to 0402 components in their tape. You need to peel-off the see-through film to access the components





> **If you're going to learn to surface-mount solder,** you need a board to practice on

Above
You don't need to go small straight away. 1206 components are relatively easy to work with

easier to use, as are those that give you enough space to work below the lens. You don't generally need particularly high magnification.

If you have a microscope that's not stereoscopic or doesn't have space underneath to solder, this can still be useful for checking your joints. If you don't want to spend a lot of money on a fancy magnification setup for soldering, a cheap USB microscope for inspecting your work can be a great asset.

COMPONENT VALUES

If you look very closely at surface-mount resistors and capacitors, you'll see some numbers on them which correspond to the values.

Usually, resistors have three numbers. The first two are digits, and the third is the power of ten that these two should be raised to to give the resistance in ohms.

Capacitors also usually have three digits which, again, are two digits and a power of ten, but this time the units are pico-farads.

You can always check the component with a multimeter if you're unsure.

Diodes and LEDs don't typically have a number on them, but do mark their polarity either with a dot on the negative connection or a |- symbol with the vertical bar on the positive connection and the horizontal one on the negative one. That said, we'd recommend always checking diodes as this isn't always the case. A multimeter should have a diode test function, or you can give them a bit of power via a resistor.

FIRST STEPS

If you're going to learn to surface-mount solder, you need a board to practice on. There are fewer options than with through-hole.

We'd recommend getting one of the boards as shown in **Figure 1**, overleaf. They're widely available from direct-from-China websites for about £5. There are loads of components, and very little in the way of instruction.

There are two parts to the kit. On either side, there are columns where you can solder multiples of the same component in place in decreasing size. At the bottom of each column, there are test pads where you can connect your multimeter to see if all the components are soldered correctly. You should see the resistance or capacitance equivalent to all the components in series, but exactly what this is will depend on what specific values your kit was shipped with.

This goes down to 0402, which is very small. This author can only get the final column working when using a fancy Metcal soldering iron or magnification, so by all means have a go, but don't worry too much if you can't get all the way down to this on your first attempt.

In the middle, there's a circuit that, if supplied with 5V, will flash LEDs in sequence. To be honest, it's quite a challenge to get this fully working, not least because there's very little detail on what components should go where (you typically get a schematic which does have details). If you're feeling determined, you can get the whole thing working, but if you're not, you can use this to practice soldering rather than trying to get everything working.

The first components we'll look at are two-pad components – resistors, capacitors, LEDs, etc. They all solder the same way. If you're using the recommended board (or similar), the largest components are 1206. If you're using a different board, we'd strongly recommend one with 0805 or larger, as it's easier to see what's going on at this size before moving down to smaller options.

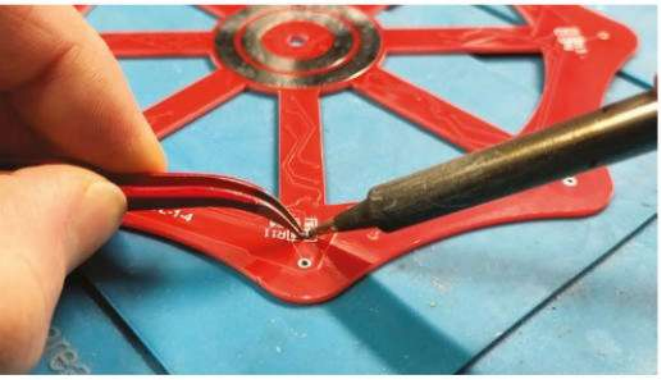
GETTING STARTED

The first step is to add a tiny bit of solder to one (and only one) of the pads for that component. Our basic technique is usually to try to add as little as possible, and that usually ends up being about the right amount (and sometimes even a bit more than ideal). To do this, put your soldering iron on the pad for a second or two, then just tap the solder on the pad. There should be a light puff of smoke, and you should have a dome of solder on the pad.





Above
You only need the slightest bit of solder on the pad – it's not there to fill a gap, just make an electrical connection with the resistor



Above
You don't need any additional solder when adding the component. Just heat the existing solder and put the component in place

Now pick up the component with your tweezers, and make sure it's oriented correctly for the PCB. Hold it in place with the tweezers while you melt the solder that you just applied to the pad.

Unlike through-hole soldering, there's no gap for the solder to fill – it's just there to make the electrical connection and hold the component in place, so you really do just need the tiniest amount. Once you go below 1206, or maybe 0805, you'll probably find it impossible to not apply too much solder, so it's a case of just doing the best you can.

If you do end up with far too much solder, you can use desoldering wick to remove some of it.

Once this has melted, make sure the component is in place, remove the soldering iron and wait for the solder to cool. This should only take a few seconds. Once it has, the component should be held in place by the solder on one of the pads.



Left
Make sure you're happy with the placement before soldering the second joint

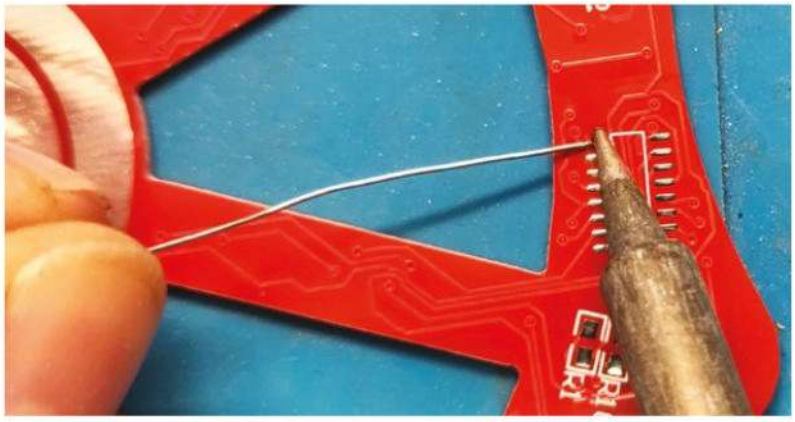


Above
It's not quite perfectly aligned, but the electrons won't mind

Make sure you're happy with the positioning of it. At this point, it's very easy to remelt the solder and adjust it, but once you've soldered the second pad, it becomes difficult to adjust.

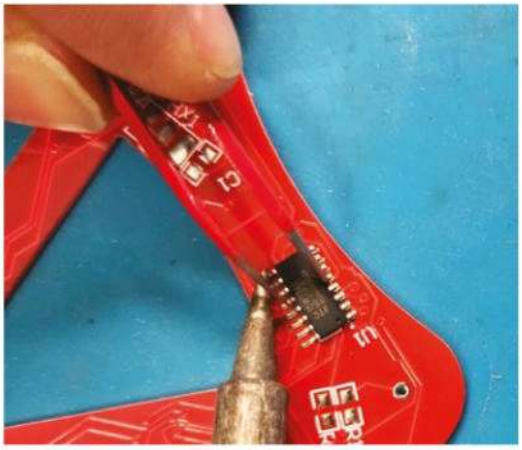
When you're happy with it, solder the second pad. Put the soldering iron on the joint, and again you want to get just the smallest amount of solder into the joint.

Take a close look at the joint – use magnification if you want – and make sure that on both sides the solder adheres to both the PCB and the component. →



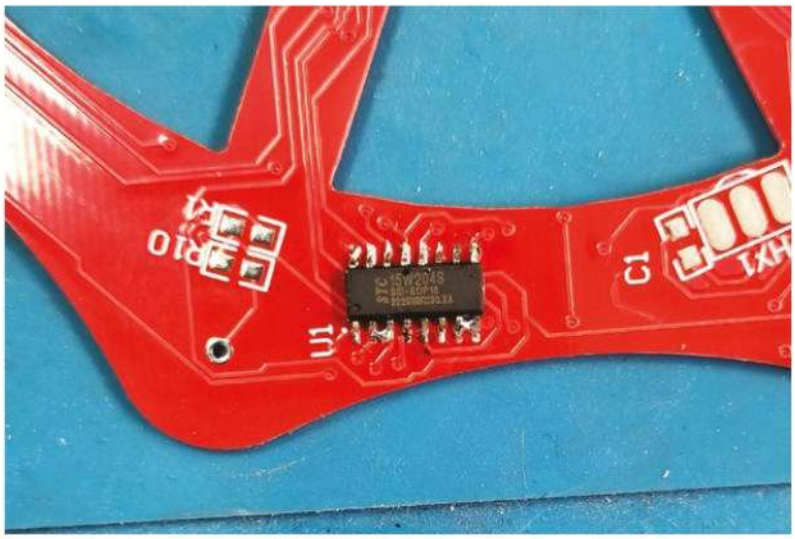
Above
These small pads need just the tiniest dab of solder

> Once you're happy with the position, **solder the diagonally opposite pin in place**



Right
The first pad will hold the component in place while you tin the opposite pad

Below
There's a good chance you'll end up with bridged pins





LOTS OF LEGS

Now we've tried a component with two pads, let's try one with lots of legs. This could be a transistor with three, or a chip with 40; the process is exactly the same. There isn't a lot of difference between doing this and the previous component, but there are more potential issues we can hit along the way, so let's take a look.

As with the two-pad component, start by putting a tiny bit of solder on one of the pads. Obviously there are lots more pads to choose from here. Ideally, you should start with the pad at the end of the row.

Now pick up your chip with your tweezers and put it in the correct place on the board. Use your soldering iron to melt the solder, and hold it in place while it cools.

Chips are much more sensitive to location than two-pad parts, so it's quite likely that you won't have got it right on the first try. If it's a little askew, you might just be able to push it into place (the leg will bend slightly), but you might also have to melt the solder with the soldering iron and reposition it. Do this as many times as needed before the chip is in the right place.

Once you're happy with the position, solder the diagonally opposite pin in place – this will hold the chip securely while you solder the rest.

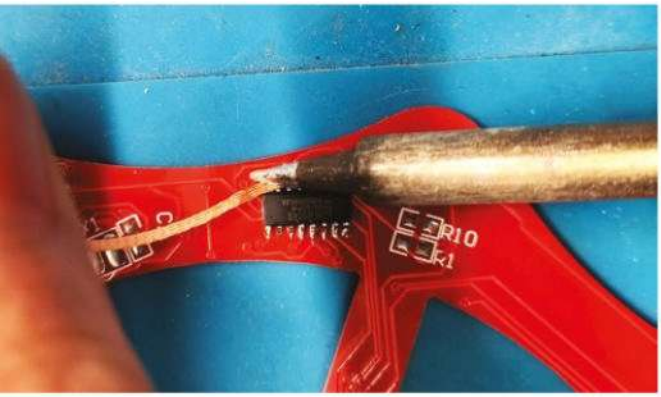
Now you can solder the pins in place. There are two techniques you can try here:

1) Apply plenty of solder across all the pins in a row. You'll end up with big blobs of solder and bridged pins. You can then use desoldering wick to remove excess solder. Place the wick across the top of the pins, then put the soldering iron on the wick (it might help to lay the soldering iron on its side so that more of the tip is in contact with the wick). Leave it for a few seconds to melt the solder, and the wick will suck up some of the excess. Remove the wick before the solder cools. You may have to do this a few times depending on just how generous you were with the solder.

2) Solder each pin individually with just the slightest amount of solder.

In either case, you might end up with a situation where there's a bridge but not very much excess solder. In this case, apply flux and heat up that solder – it will usually unbridge itself.

Option 1 is often easier and more likely to result in properly soldered joints, but depending on your iron and solder, option 2 can be a bit quicker.



Above
A bit of wick will remove any excess solder. If this doesn't work, try adding flux and reheating

It can be hard to ensure that every pin is soldered down, so again, it's worth looking over with magnification if possible.

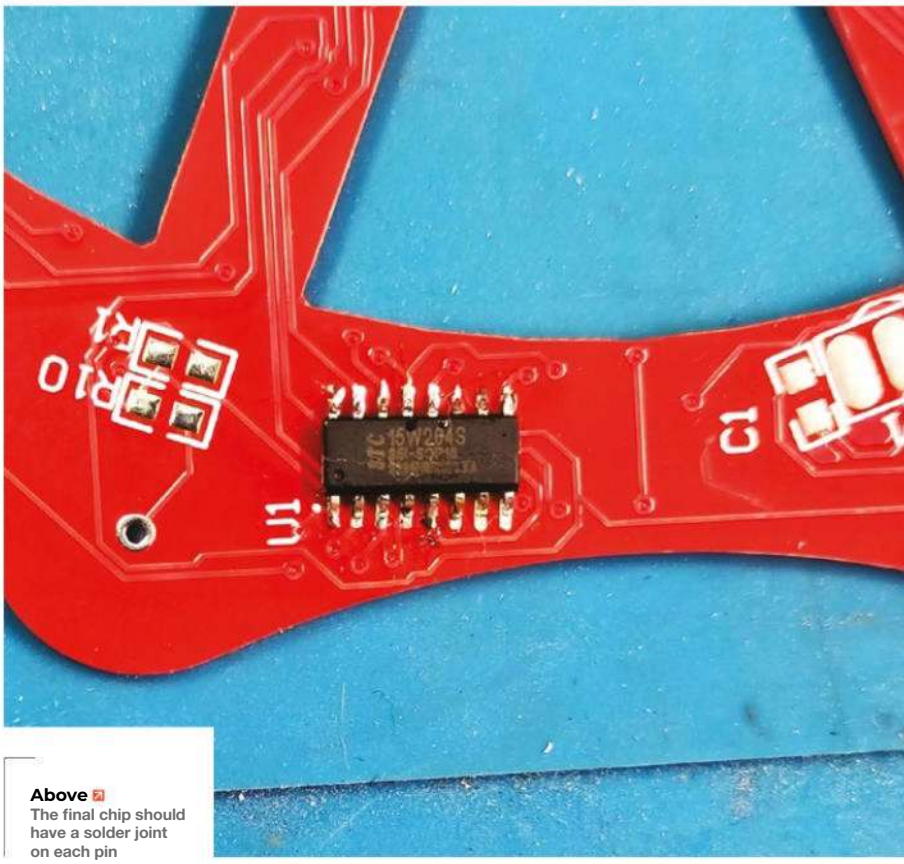
GETTING SMALLER

Surface-mount soldering isn't necessarily harder than through-hole soldering, but things can get small. Very, very small. As you get smaller, the technique doesn't change, but it does get more difficult.

There are a few things to consider. In no particular order, here are our tips for soldering small surface-mount components:

- Using thin solder wire will make it easier to apply the correct amount of solder. We use 0.5 mm wire, but you can get thinner wire if you're going to do a lot of fine-pitch soldering.
- Having a fine tip on your soldering iron can help, but only if your soldering iron can correctly deliver heat into it. We've found that many soldering irons can't hold a fine tip at the correct temperature, and the very point ends up colder than slightly further down the nose. You might be better off with a blunter tip, but with the full temperature on the end.
- Heat travels down PCB traces. You can place your soldering iron on the trace that leads up to a component rather than poking it into the component itself.
- Flux. If you're having problems, add flux.

No matter how skilled you get, soldering small components is always a time-consuming process. Next month, we're going to look at using heat guns, hot plates, and PCB ovens to populate PCBs without having to poke a hot stick at each individual pad.



Above
The final chip should have a solder joint on each pin

FLUX OR NO FLUX?

We'd recommend using flux-cored solder, and as such, you don't necessarily need to add additional flux. However, it can make things easier. It can make things a lot easier.

It really comes down to personal preference – we solder larger SMD components without flux, but use flux when things get small.

Flux comes in a few forms, but most commonly, paste, liquid, or pen. Pens can just draw flux directly onto the PCB. Liquid usually has a fine nozzle for dispensing it. Paste can either be in a tub where you'll need a matchstick or similar to dab some onto the PCB, or in a syringe that lets you drop a bit on. Paste flux becomes a liquid when heated, so don't add too much or it'll spread out across the board.

There are two ways of using flux. One is before you've started. In this case, you put a little flux on the exposed pads on the PCB before starting and then continue as usual. The second is if something has gone wrong – such as bridged contacts. A dab of flux and a reheat can work wonders.

Flux will burn off, and there will be more smoke than if not using flux, so make sure you're ready for this (ideally with fume extraction).

Different fluxes have varying levels of toxicity, but as a general rule, it's best to avoid getting it on your skin, and you definitely want to wash any off that does get on your skin well before touching your face or eating. Take a look at your flux's material safety data sheet for more information.

Traditionally, fluxes have been washed off with isopropyl alcohol (IPA) after the board is finished to stop the acid in the flux from damaging them. This isn't strictly necessary with many modern fluxes, but it's not a bad idea to remove any residue. There are water-washable fluxes available if you would rather not use IPA.

TUTORIAL

Part 01

Make a Tide Tracker with APIs and an e-ink display

Surf's up, or is it? Save yourself a wasted trip to the beach with our handy tide tracker



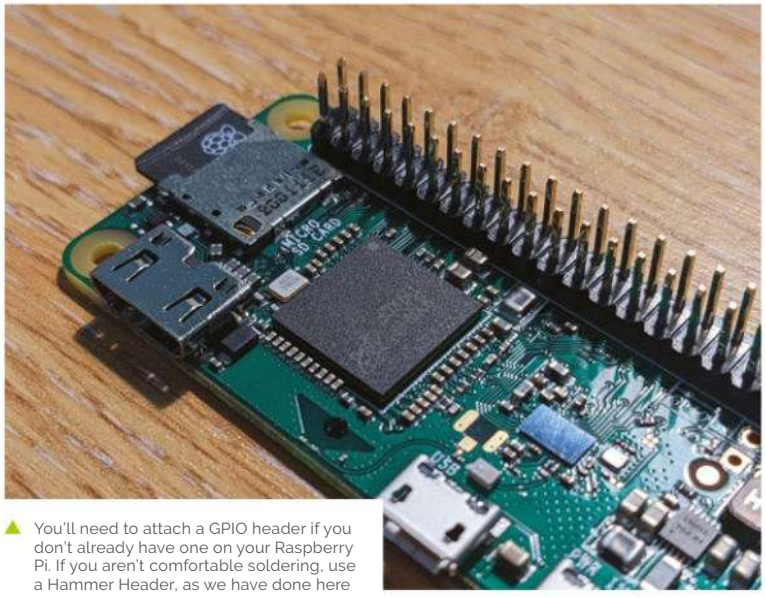
MAKER

Nik Rawlinson

Coffee-drinking typewriter lover with a penchant for pencils and paint. Frequently found staring at the sea from the back of a camper van.

nikrawlinson.com

Tides are predictable, but it's still easy to lose track of the times for high and low water. As a result, you might pull up at high tide to find the beach has disappeared… or at low tide, and found your boat grounded. Put an end to that once and for all with our handy tide tracker, which uses an API to download tidal forecasts for more than 600 monitoring stations around the UK. The results shouldn't be used for navigation or other water-borne activities, but they may be just what you need to save yourself a wasted journey, or to keep an eye on the tide cycles however far you live from the coast.



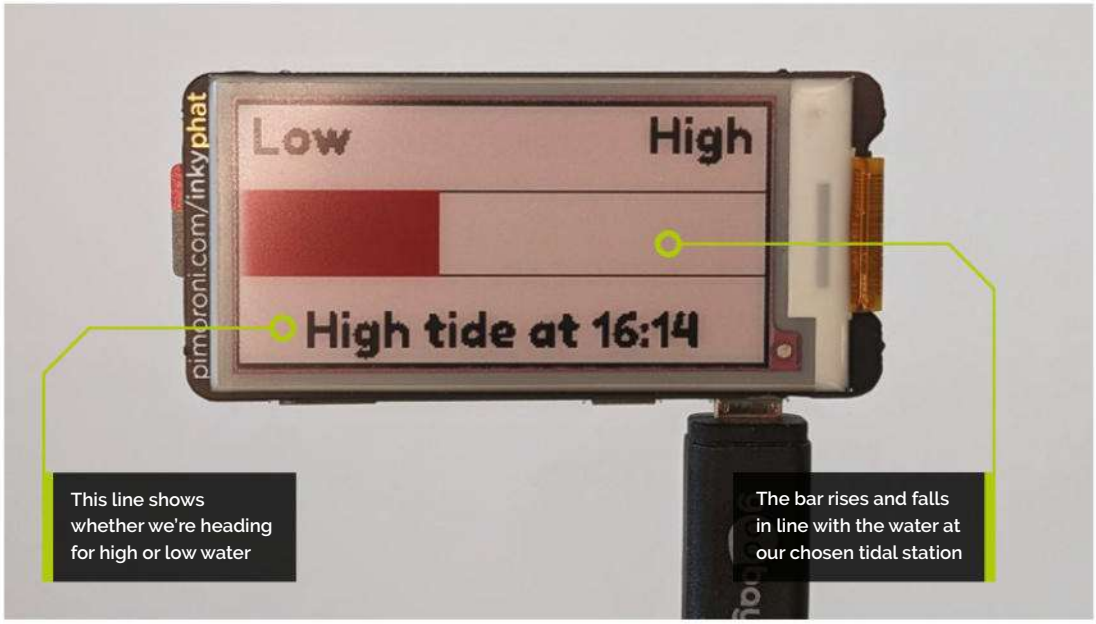
You'll need to attach a GPIO header if you don't already have one on your Raspberry Pi. If you aren't comfortable soldering, use a Hammer Header, as we have done here

01 Attach a header

We're using a Raspberry Pi Zero 2 W for this project. If you're using the same and don't already have a GPIO header in place, you'll need to solder one using the two rows of connectors at the top edge of the board. If you're not comfortable soldering, instead use a GPIO Hammer Header, which you can knock into position using a hammer and installation jig. Once your headers are fixed, attach the e-ink display, then connect your Raspberry Pi to a USB power source and allow it to boot

02 Install display libraries

Whatever display you're using, you'll first have to install the libraries that Python needs when it wants to communicate with it. We're using an Inky pHAT, for which Pimoroni has published an excellent getting started guide at **magpi.cc/getstartedinkyphat**. Follow the instructions there for installing the necessary software, by opening a new Terminal window (click the icon on the taskbar) and typing:




```
curl https://get.pimoroni.com/inky | bash
```

When the process has finished, reboot your Raspberry Pi.

03 Register for an Admiralty account

We're going to use data compiled by the UK Hydrographic Office. Although it charges for much of its data, its free Discovery tier gives access to data for 607 tidal stations around the UK. Open a browser window at **magpi.cc/uktidalapi**, check the Terms of Use and, if you're happy to be bound by them, set up an account by subscribing to the service. When your account is up and running, log in, then click your name at the top right of the page, followed by Profile. Click Show on the line beside 'Primary key', and copy the subscription key this reveals into a plain text document for later use.

04 Find your tidal station

You now need to find the numeric identifier of the tidal station whose data you want to retrieve. While still logged in, point your browser at **magpi.cc/tidalstations** and type the name of a coastal location in the box labelled 'name'. Leave everything else on the page as it is, and click the Send button. Assuming you entered a valid location, the query's output will appear in the 'Response content' box. Copy the variable beside Id in the Properties section. We searched for Broadstairs, for which the 'Id' variable reads 0102A. You'll need to include the station identifier in your query.

05 Understand how tides work

Our gauge will illustrate the tide's rise and fall at any station, always showing how far it is from its maximum or minimum height. We don't need live height data to produce the graph if we apply the rule of twelfths, which governs how

> "Our gauge will illustrate the tide's rise and fall at any station"

tides move and gives a good approximation of the current position based on the interval between the most recent and next low or high tide. If you break down the interval into six equal periods, the tide height will change by 1/12th of its total range in the first and last periods, by 2/12th (a sixth) of its range in the second and fifth periods, and by 3/12th (a quarter) of its range in the third and fourth periods. You'll apply these ratios in the code.

06 Get coding

Type out or download the **tides.py** code and amend it to your requirements, changing lines 3 to 14 as appropriate if you're using a different e-ink display. Replace `<id>` on line 19 with the ID of the tidal station you're monitoring, and `<key>` with

You'll Need

- Inky pHAT **magpi.cc/inkyphat**
- GPIO headers **magpi.cc/colourheader**
- (Alternatively) GPIO Hammer Header **magpi.cc/hammerheader**

THE MAGPI



This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at **magpi.cc**

Top Tip

Not in the UK?

The data we're using only covers UK tidal stations, but equivalent data is routinely published by government agencies around the world.

your subscription key. Lines 23 to 33 are worth closer examination. Because the query pulls down every low or high tide event for 48 hours, you need to find when the next one is due. So, the code first cycles through each event in the returned data and converts its time stamp to its UNIX equivalent, which counts the number of seconds since 1 January 1970.

07 Deal with the unexpected

Notice how line 26 reads:

```
at = at[:19]
```

The variable `at` holds the time stamp associated with each high or low tide event. Usually this takes the 19-character form YYYY-MM-DDTHH:MM:SS (as you'll see, there's a T in the middle, which marks the transition from date to time). However, occasionally the time stamp returned by the query is more specific, and includes an additional half second. We need to discard this in our calculations, so line 26 trims the contents of the `at` variable to 19 characters before moving on. If there was no additional half second, this would have no effect, but also wouldn't throw up an error.

08 Calculate the tidal cycle

As the code continues cycling through the data, it stores the time of each tidal event until it reaches the first one that hasn't yet happened, then quits the loop. It now has two time stamps stored in variables: one for the previous event, and one for the next. Subtracting the former from the latter tells us how long, in minutes, the transition from low to high or high to low tide takes. We divide 100 by that number to work out what proportion, in percentage terms, is represented by a single minute. Then, in lines 54 to 71, we set a multiplier, based on our rule of twelfths, that will allow us to plot a fair approximation of the tide's position within its cycle on our display.

09 Define the bar chart

The tide's progress will be tracked using a bar chart, the length of which is calculated on lines 82 to 84. However, it wouldn't be clear from just a glance at the bar whether the tide was rising or falling. So, it's accompanied by a line of text indicating what the next event is (high or low tide) and when it will occur. For the latter part of the text (the bit dealing with the time), we'll extract a subsection of the data held in the variable `at`. On line 86, we assign the 11th to 16th characters of `at`, which contains the two hour digits, a colon, and two minute digits, to a new variable called `event_time`.

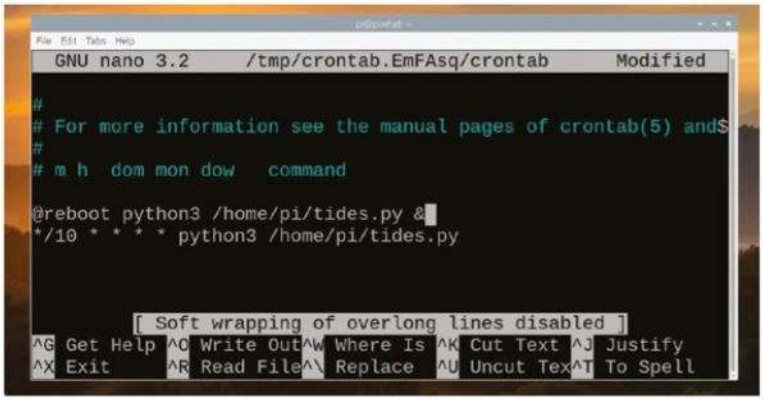


We're using an Inky pHAT display, but you can adapt the code for other screens

10 Refresh the display

Now that our code has constructed the display's contents in memory, we need to transfer it to the e-ink panel. This task is handled by lines 106 and 107, which send the constructed image to the screen when we run the code in step 11. If you're using a colour display, you may notice that it will take a few seconds to refresh. This is normal. However, if you want to speed it up, go back into your code and change line 9 to:

```
inky_display = InkyPHAT("black")
```



Use cron to run your code periodically and keep the reading updated

tides.py

DOWNLOAD THE FULL CODE:
magpi.cc/tidespy

Language: **Python**

```
001. # import necessary libraries for data retrieval and drawing to Inky pHAT
002. import urllib.request, json, datetime, time
003. from PIL import Image, ImageFont, ImageDraw
004. from inky import InkyPHAT
005. from font_fredoka_one import FredokaOne
006.
007. # define Inky pHAT parameters
008. # change "red" in line below as appropriate for your display
009. inky_display = InkyPHAT("red")
010. inky_display.set_border(inky_display.BLACK)
011. img = Image.new("P", (inky_display.WIDTH, inky_display.HEIGHT))
012. draw = ImageDraw.Draw(img)
013. width = inky_display.WIDTH
014. percentwidth = 100 / width
015.
016. # draw down data from the UK Hydrographic Office
017. # replace <id> with your station id
018. # replace <key> with your subscription key
019. with urllib.request.urlopen("https://admiraltyapi.azure-api.net/uktidalapi/api/V1/Stations/<id>/TidalEvents?duration=2&key=<key>") as url:
020.     data = json.load(url)
021.     datalength = len(data)
022.     # cycle through data to find next event
023.     for x in data:
024.         event = (x['EventType'])
025.         at = (x['DateTime'])
026.         at = at[:19]
027.         when = datetime.datetime.strptime(at,"%Y-%m-%dT%H:%M:%S")
028.         unix = datetime.datetime.timestamp(when)
029.         now = time.time()
030.         if unix > now:
031.             break
032.         previous_event = event
033.         previous_at = unix
034.
035. # calculate minutes until next event
036. next = (unix - now) / 60
037.
038. # calculate minutes since previous event
039. previous = (now - previous_at) / 60
040.
041. # calculate time between events
042. total = next + previous
043.
044. # calculate what percentage is represented by a single minute
045. percent = 100 / total
046.
047. # multiply percentage by time since last event to find elapsed percentage
048. progress = percent * previous
049.
050. if event == "LowWater":
051.     progress = 100 - progress
052.
053. # calculate start point and set multiplier
054. if progress < 8.33:
055.     start = 0
056.     multiplier = 1
057. elif progress > 8.33 and progress < 25:
058.     start = 8.33
059.     multiplier = 2
060. elif progress > 25 and progress < 50:
061.     start = 25
062.     multiplier = 3
063. elif progress > 50 and progress < 75:
064.     start = 50
065.     multiplier = 3
066. elif progress > 75 and progress < 91.66:
067.     start = 75
068.     multiplier = 2
069. else:
070.     start = 91.66
071.     multiplier = 1
072.
073. #subtract completed start point from minutes remaining to find remainder
074. remainder = progress - start
075.
076. #calculate percent value of remaining minutes
077. remainder_percent = percent * remainder
```

DOWNLOAD THE FULL CODE:
magpi.cc/tidespy

```
#multiply remaining minutes by multiplier
remainder_percent = remainder_percent * multiplier

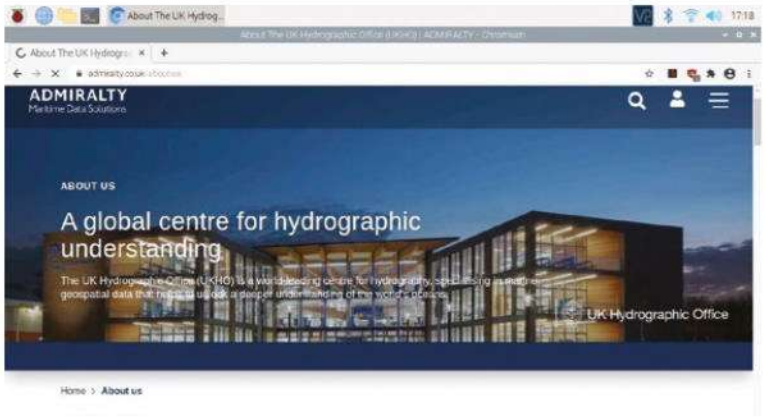
calculated = remainder_percent + start
bar = calculated
barout = (width / 100) * calculated

event_time = at[11:16]

if event == "LowWater":
    e = "Low tide at " + event_time
else:
    e = "High tide at " + event_time

mid = inky_display.HEIGHT / 2
font = ImageFont.truetype(FredokaOne, 20)
low = "Low"
high = "High"
w, h = font.getsize(high)
w1, h1 = font.getsize(e)

draw.line(((0,inky_display.HEIGHT * .33), (inky_display.WIDTH, inky_display.HEIGHT * .33)), fill = 1, width = 1)
draw.line(((0,inky_display.HEIGHT * .66), (inky_display.WIDTH, inky_display.HEIGHT * .66)), fill = 1, width = 1)
draw.rectangle(((0,inky_display.HEIGHT * .33), (barout,inky_display.HEIGHT * .66)), fill = 2, width = 1)
draw.text((5,0), low, inky_display.BLACK, font)
draw.text((width - 5 - w, 0), high, inky_display.BLACK, font)
draw.text(((width - w1) / 2, inky_display.HEIGHT - h1), e, inky_display.BLACK, font)
inky_display.set_image(img)
inky_display.show()
```



The data we're using is compiled by the UK Hydrographic Office

11 Run your code

If necessary, move the code into the root of your user folder and save it as **tides.py**. Now, open a Terminal window by clicking the icon on the taskbar. At the prompt, type:

```
python3 tides.py
```

...and press **RETURN**. The Python interpreter will run your code and, after a few seconds, you should see the display refresh to show the current tidal position, with details of the next tidal event on the line below.

12 Run your code automatically

Assuming your code did what you expected, you can set it to run automatically and refresh at set intervals. Type:

```
crontab -e
```

If you've never edited crontab before, you'll be asked which tool you want to use. Select 1 for Nano. When Nano opens, key down to the bottom of the file, create a new line, and type:

```
@reboot python3 /home/pi/tides.py &
*/10 * * * * python3 /home/pi/tides.py
```

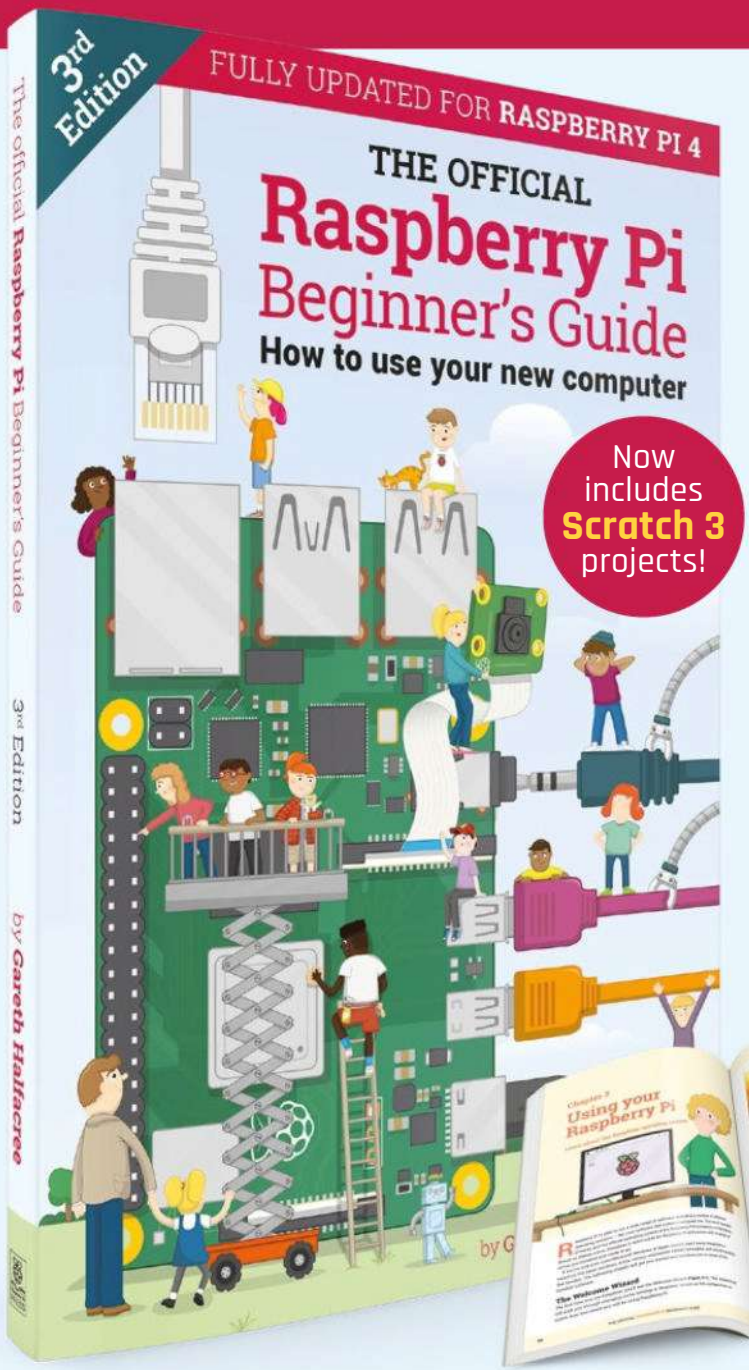
The first line tells Raspberry Pi OS to run your code as soon as it starts, with the **&** at the end telling it to run the process in the background. The second line tells it to run the code every 10 minutes thereafter. Press **CTRL+X** to quit Nano, confirm that you want to save your edit, then reboot your Raspberry Pi.

Top Tip

Next issue

In the next issue, we'll rework this project using MicroPython, for use with the colourful Galactic Unicorn with built-in Raspberry Pi Pico W.

THE OFFICIAL Raspberry Pi Beginner's Guide



The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

Plus much, much more!

£10 with FREE worldwide delivery



Buy online: magpi.cc/BGbook

TUTORIAL

Supercapacitor free flight planes

Take to the skies with a minimalist flying machine



Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

A **free flight rubber band-powered plane is almost a rite of passage for those interested in making things that fly.** You can get pretty decent flight times from very affordable, small, and simple designs, and it's a great way to learn the art of 'trimming', tweaking, and tinkering to get the flight characteristics you want. For this 'powered' free flight, there are other alternatives, but electric free flight is a nice, affordable, quiet and cheap-to-run hobby. There are lots of designs out there to build electric planes that carry a small LiPo cell and a timer circuit, so that the propeller is only driven for a short and specific amount of time – often, less than a minute. This is an excellent approach, but there's a little bit of a weight penalty carrying a battery and also a bit of complexity in building a timer circuit. The electric option we're going to look at is using a supercapacitor to drive a motor and propeller, adjusting the amount you charge the supercapacitor to change the driven flight time.

You can actually buy small foam gliders with motors, propellers, and supercapacitors fitted. Whilst we are sure many of these are excellent, we were introduced to supercapacitor flight as a member of a local indoor flying club had bought some of these for resale. On arrival, all the foam components were damaged, the member was refunded, and they presented us each with some of the below par kits



Right
Although really these are better for indoor flying, on a very still, calm day you can venture outdoors with your flying creation!

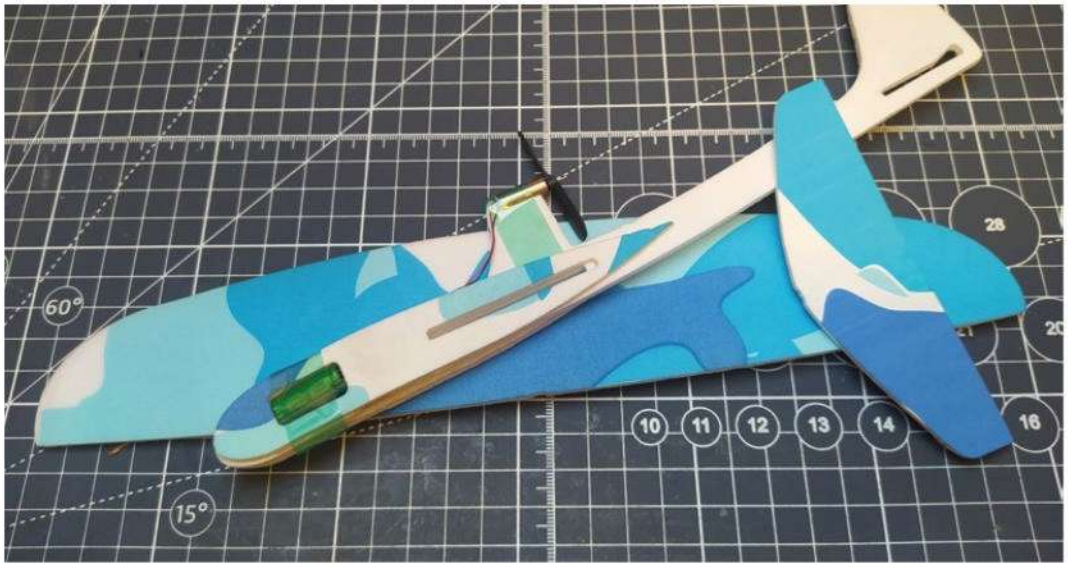


Figure 1
These foam kits are widely available but often arrive damaged – they are then useful as a source of spare parts!

for us to pillage the drive systems from. This, in turn, led us into exploring DIY supercapacitor setups made from components, and building, flying, crashing, and rebuilding lots of test aircraft!

The donated foam kits, **Figure 1**, have a small 6 × 20 mm brushed motor set up in a pusher formation, where the motor and prop are pointing towards the rear of the aircraft. The system 'pushes' the air back over the rear of the aircraft, creating a forward thrust. This is a nice approach for a variety of reasons; including that, if your aircraft crashes, the motor and propeller aren't the first thing to hit the ground and are a little protected. Looking at this setup from the rear, the motor is spinning counterclockwise and the propeller blades are angled to work in this

> **The system 'pushes' the air back over the rear of the** aircraft, creating a forward thrust

direction of rotation. The donor foam kits are fitted out with a 5-farad supercapacitor, and this is mounted onto a PCB which has a tiny barrel jack connector. The idea is that you connect a battery pack to the system with a barrel jack and this charges the capacitor. On the donor system, the PCB has a surface-mount LED that indicates the level of charge, increasing in brightness as the supercapacitor charges. As soon as you disconnect the barrel jack, the supercapacitor starts to discharge through the motor and the propeller starts spinning. It will run with some gusto for around 30–40 seconds and will continue to spin at lower power for a couple of minutes.

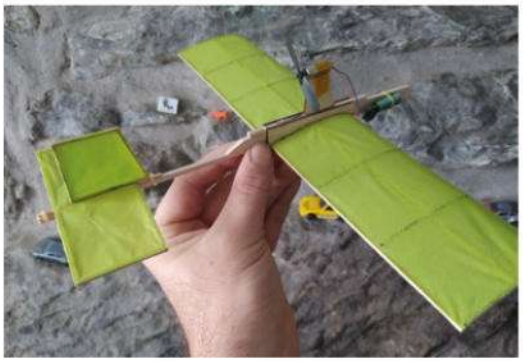


Figure 2
Our first attempt at a DIY supercapacitor plane was this balsa and tissue design using the running gear salvaged from a foam model

The majority of supercapacitors that get used for free flight model flying are rated at 2.7 V and, indeed, the ones attached to the donor foam gliders are such. Therefore, we were a bit surprised that the kits came with a 4 × AA battery box, as this provides far too high a voltage. In fact, we sacrificed one kit trying it out and it definitely caused the supercapacitor to lose capacity quite quickly. It's more common for people to charge these using 2 × AA batteries in a battery box, and this has worked well for these donor supercapacitor setups, as well as our later DIY versions. If you get really fancy, you can develop a constant current charge system to really dial in accurate charges, but we've found two AAs to be perfectly adequate.

For our first attempt, we used the salvaged supercapacitor setup from the foam kit and decided that it would be a sensible idea to try and make the build roughly similar in terms of weight and dimensions (**Figure 2**). We decided to go with a lightweight balsa frame covered with tissue paper which is a traditional way of aeromodelling. Aiming for 9.5 grams, we made two wing sections that were around 7 cm by 12 cm in size, and attached them →

YOU'LL NEED

- **8 × 20 mm counterclockwise brushless motor**
- **6.8–10 F supercapacitor**
- **Some thin, flexible wire and soldering equipment**
- **Thin, lightweight foam**
- **Paper drinking straws**
- **Glues and tapes**
- **One 2 × AA battery holder**

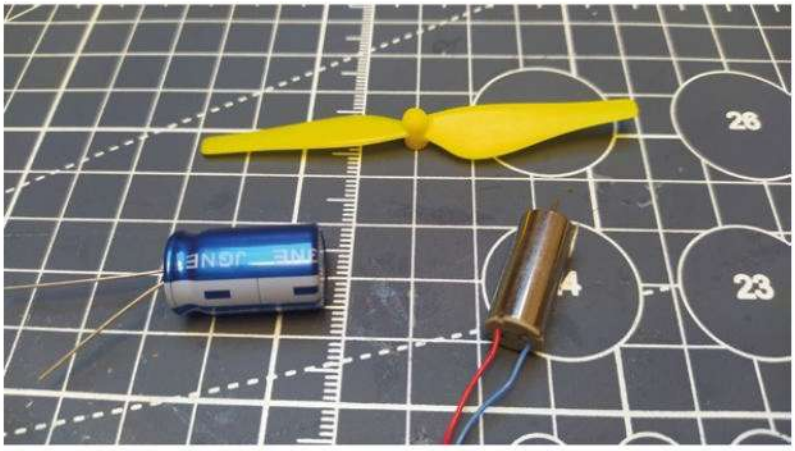


Figure 3
A larger supercapacitor, an 8 × 20mm motor, and a larger propeller for our DIY power pod build



Figure 4
The start point of a power pod is to cut a section in the straw to mount the motor in

to a U-section that slotted over a 3mm wide balsa fuselage section. We similarly made a tail section with a horizontal stabiliser, and a vertical surface that had a small hinged section to act as a rudder. As this was an experiment, we made the flight surfaces, the wing and the tail, so that they were held on by tiny rubber bands. Making it so that we can move and reposition the components on the aircraft is a great idea when you are working with a new design and want to try out ideas. We tend to err away from the massively complex fluid dynamics engineering of plane design and used some rules of thumb, such as making the tail horizontal surface around 20% of the surface area of the main wing. We cut a small balsa 'pylon' which lifts the motor and prop so that it is clear of the fuselage. We tacked the motor onto the pylon with a spot of superglue, but we attached the PCB and supercapacitor with a rubber band so we could again move this up and down the nose end of the fuselage to move our centre of gravity a little.

Testing the little model revealed that we had designed the wing with too much under-camber and perhaps needed more of an angle of incidence (the angle where the wing points upwards a little at the front when compared to the flat rear stabiliser). It also perhaps had a little too much weight – this meant that we could get reasonable flight characteristics from it whilst the motor was creating some thrust, but it would fall out of the sky when the air speed slowed. That said, we managed to get it trimmed to the point where it would fly a large diameter lap of a gymnasium smoothly and then glide into land. It was certainly encouraging enough to make us explore more.

For our DIY build, we needed to come up with a way to charge a supercapacitor and a way to switch the motor system on and off, whilst not adding masses of weight. We decided to take a step up slightly in size of motor and supercapacitor to give us a little more thrust to play with. We ordered some 8 × 20 counterclockwise (CCW) brushless motors and a couple of different capacity supercapacitors, 6.8F and 10F (**Figure 3**). We wanted to stick with a pusher-type approach and so we used some larger 80mm propellers which are some spares that we had for our small Ryze Tello quadcopter. You can buy sets of these online quite cheaply, and they fit directly onto the shafts of the 8 × 20mm motors. You get different propellers in a quadcopter set, as different corners of a quadcopter rotate differently, but you are looking for the CCW-marked propellers within the set of four.

> **"We tend to err away from the massively complex** fluid dynamics engineering of plane design**"**

We've made numerous test aircraft using this same setup where all the supercapacitor, motor, charge points, and wiring are attached to a paper drinking straw that acts as the plane fuselage. It's a handy system that could be transformed to a variety of different plane experiments and so we're calling it a 'power pod'. The idea is straightforward. At one end of the paper straw is the motor. We've begun by cutting away a section of the straw about 15mm long, but leaving a part of the straw that we can add a dab of glue to and mount the motor (**Figure 4**).



Figure 5
Mounting the supercapacitor is simply done with a rubber band, allowing for it to be repositioned later

QUICK TIP

These small brushless motors with small propellers attached spin incredibly quickly and they can really hurt your fingers. Be careful that you aren't in their way!

We've used a small dot of superglue to attach the motor with the back of the motor case pushed up flush to the straw. Don't worry if this feels a little flimsy at this point, as we usually reinforce this section when we attach it to a plane. At the other end of the straw, we tend to attach one of our larger supercapacitors, non-permanently with a rubber band (**Figure 5**).

Somewhere towards the middle of the straw, we added two small sections of sticky-back copper tape. We've experimented before with this tape, and enjoy cutting conductive parts from it, either by hand or using a vinyl cutter. Our tabs are around 6 mm wide and about 12–15 mm long. Make sure that each tab is on opposite sides of the straw and that they are not contacting each other, as these copper tabs are going to be our charge points (**Figure 6**).

Next, find a small plastic straw where the hole is around 1 mm diameter. A good option for these is the types of tubes that come with cans of spray oil, such as many types of general lubricant. Another source is some soft drinks cartons that use similar small-bore straws. Cut around 15 mm of the straw off and then cut a small piece of stiff wire; we've used MIG welding wire, but a guitar string would work similarly. Bend around 5–7 mm of the wire into a right angle and slide it into the straw section you have just cut. When fully inserted, you should have 3–5 mm of wire poking out of the other end of the short straw section.

Take the wire out of the small straw section and then glue the small plastic straw section onto the paper straw in the position you want the switch to be, but between the copper tabs and the motor; we've tended to use a drop of superglue for this. On the wire that you cut and bent, lightly sand the bent handle section so that it's clean enough to solder to. Cut a short length of thin, flexible wire and strip a short section of insulation off each end. Wrap one of the uninsulated ends around the stiff wire switch bent section and solder them together. Once cooled, place your switch wire into the small tube glued to the straw (**Figure 7**). Take the other end of that wire and solder it to the right-hand edge of one of the copper tabs, the end of the tab closest to the motor. This is going to be the positive side of the circuit.



Figure 6
Adhesive copper tape is used to create charge pads with connections soldered onto them

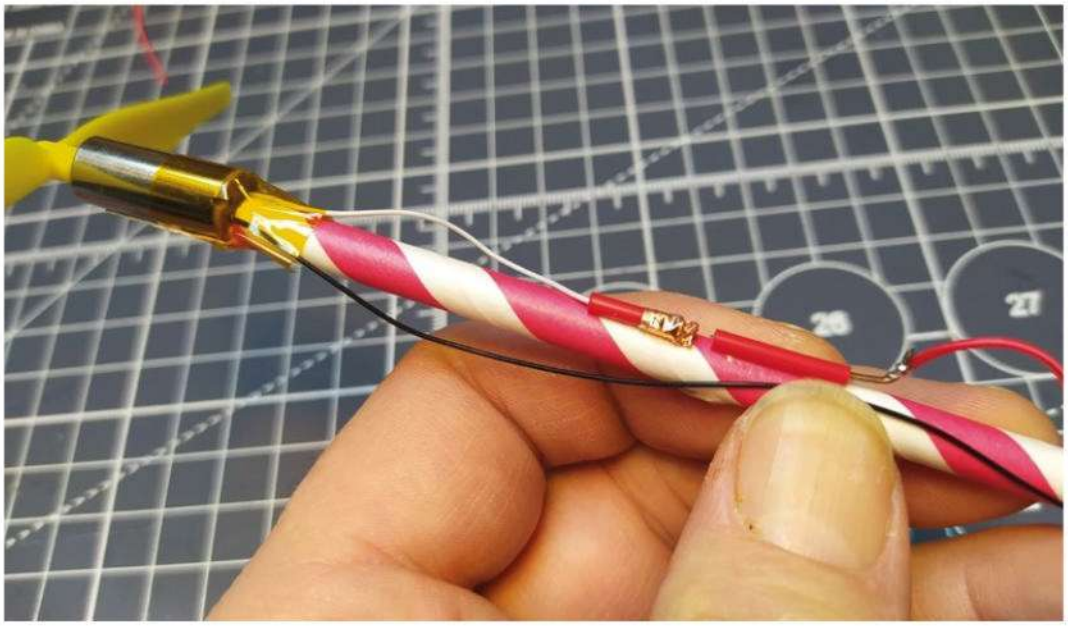


Figure 7
A simple on/off switch for the motor can be made from some MIG welding wire and a small plastic tube

QUICK TIP

We leave our supercapacitor wires longer to allow you to move it along the straw to change the balance of the aircraft. Once you're happy, you could shorten the wires to reduce weight.

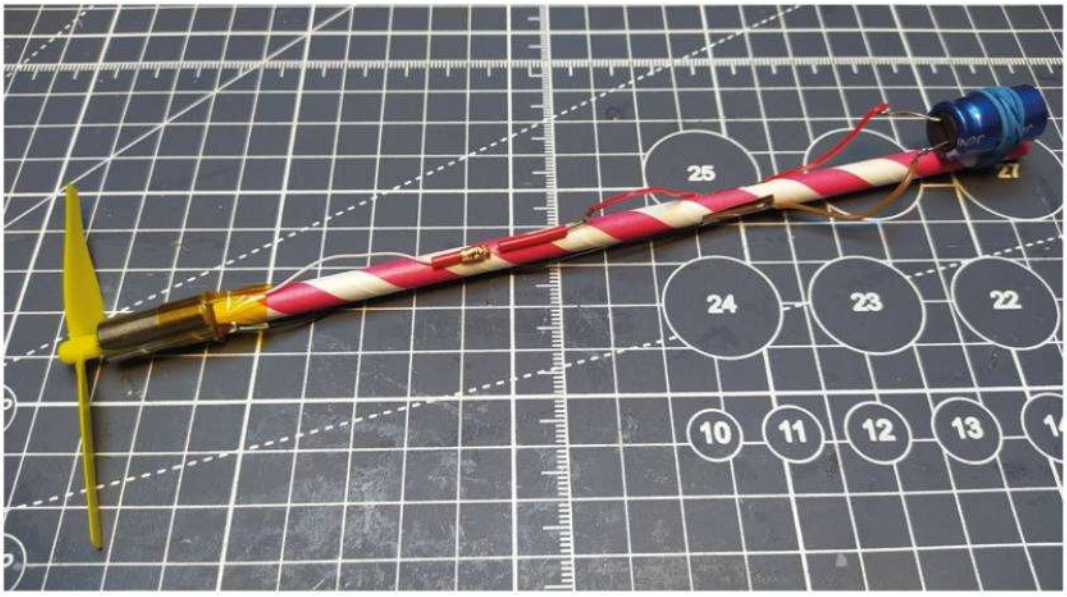


Figure 8
Our completed power pod with motor, supercapacitor, switch, and charge points

Next, solder a short flexible section of wire to the positive lead on the supercapacitor, and then solder the other end of that wire to the left-hand side of the positive copper pad that we connected the switch wire to earlier.

On the opposite side of the straw, solder a wire between the negative lead on the supercapacitor and the closest edge of the copper tab to it. Then solder the negative motor wire to the opposite end of the same copper tab. This means the motor is always connected to the negative lead of the supercapacitor.

To finish the positive side of the circuit, we are going to cut another section of the small tube/straw we used earlier. Cut around 5–6mm and thread it over the positive wire attached to the motor. Next, strip around 12mm of insulation off the end of the motor positive wire and pull the short length of tube up towards this uninsulated section. Then wrap the uninsulated section around the end of the short length of tube and stick it into place by covering it with a thin strip of the conductive copper tape. Squash this all together so that you have a small bundle of copper tape at the end of the tube which is connected electrically to the motor positive wire. Now glue the plastic tube in such a position that it is close to the other switch tube section and that, when you push the MIG wire section all the way in, it will touch the copper pad on the motor-connected side. However, you should be able to move the MIG wire section back to create a gap in the circuit (as in **Figure 8**).

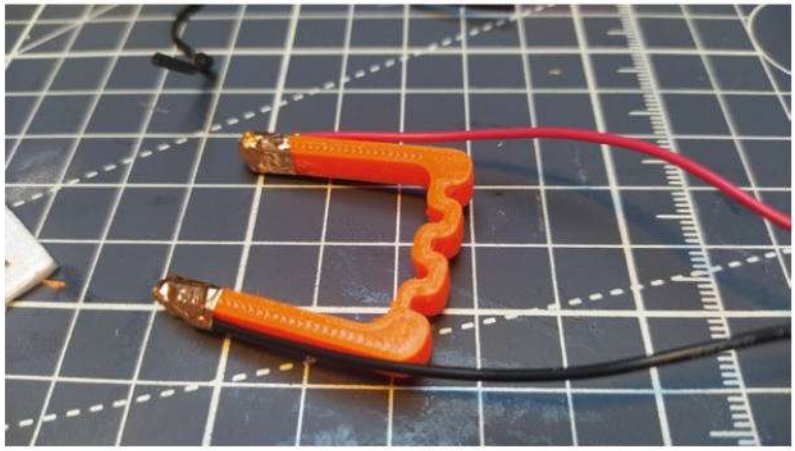
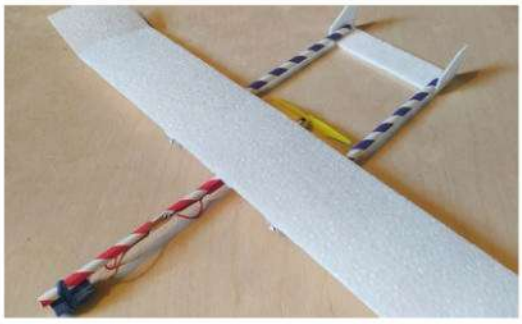


Figure 9
A 3D-printed spring clamp with positive and negative charge pads can be touched to the charging pads on the power pod

To charge our capacitor, we are going to press a positive and negative contact onto the pads on each side of the power pod assembly we've created in the previous section. We used FreeCAD to make a tiny spring clamp (**Figure 9**), but you could make something similar perhaps by using a laundry peg with some kind of limiting block. The 3D-printed clamp is available on Thingiverse (**hsmag.cc/PrintedClamp**) and it has worked really well. The clamp has two holes at the upper end of each side, and we have used the same technique as we did for the switch on board the power pod. We stripped the positive and negative leads coming out of a 2 × AA battery box, salvaged from a broken set of LED lights, inserted the stripped wire sections and wrapped them around the clamp ends. Securing this with copper tape, it creates a positive and negative clamp connection we can touch to the copper pads.

To charge the system, open the switch on the power pod and then gently press the clamp pads onto the correct pads, being really careful to check polarity! It would be easy to touch the clamp to the incorrect pads, which is unlikely to be dangerous, but isn't going to prolong the life of the supercapacitor. You can experiment with your own systems, but for our 10F supercapacitor, we find counting to around



twelve seconds gives it a good charge. You can then carefully slide the switch wire over to close the circuit and the motor should run.

We don't totally know why, but we have become obsessed with the idea of building a free flight twin-boom plane, where the motor and pusher propeller sit between the rear tail booms (**Figure 10**). It's a nice design for our power pod approach and you get a fairly straight thrust line through the fuselage as it's all essentially a straw. We haven't quite perfected the plane with each iteration (currently testing Mk4 and building Mk5 at time of writing), but we can happily share some rough dimensions and ideas.

> A lightweight bond can be made by brushing **the pieces with slightly watered-down PVA glue**

First of all, we are building as light as possible! To enable this, we are using some 2 mm foam that comes on a roll and is sold as an insulation product that can be adhered to walls under wallpaper. We've seen a couple of brands, but the most common is 'Diall' and you can buy a ten-metre roll for around £6.50. It's very light and very thin, so we have been laminating two pieces together to form our main section of the wing. As the foam is supplied as a roll, it naturally has a very shallow curve to it, which plays to our advantage when making a slightly aerofoiled wing. We've been sizing the wing at around 8 cm 'chord' length (the shorter direction) and between 28 and 36 cm wide. We cut two sections to size using a steel ruler and a sharp craft knife, and then glued these sections together. We've found that a lightweight bond can be made by brushing the pieces with slightly watered-down PVA glue, but you need to leave this to cure for a full day and, whilst light, it isn't the strongest. As a compromise, we have tended to stick the front edge of the wing and the wing tips using UHU Por, which is a foam-safe glue, and the rest of the wing surface with watered-down PVA. This combination of glues means that the wing can be handled after an hour or so and the UHU reinforces the leading edges which take some abuse on landing.

All our twin-boom design attempts so far have included adding some small dihedral tips to the wing ends. These have been between 4 and 6 cm in length, and you need to carve the connecting end slightly so that when you glue it to the wing, it makes a neat joint. We've tended to angle ours upwards at around 20 degrees, and these tips definitely create a more stable and self-correcting glide.

We've positioned the booms based on having enough space for the propeller to swing between them and also to give a reasonable tail surface area. The latest iteration uses an 80 mm propeller, and we have the booms spaced out at 10 cm. We have built up the front edge of the wing with around 5 mm of foam and paper to both lift the angle of incidence of the wing and also reinforce the design. The wing therefore sits at around a 3-degree angle compared to the horizontal tail section (**Figure 11**). This helps to generate lift, but as we say, feel free to experiment widely and share your results!

The 10 cm by 3 cm tail section is made from a single layer of the 2 mm foam, so whilst it's fragile, we can also flex it a little to bend in angles to create lift and, similarly, we can use the small foam vertical tails to create small rudder angles. We're aiming to get the aircraft to climb up in a few circles for around 15 to 20 seconds before gliding back down. We can't say we are 100 percent there yet, but it's great budget fun and a brilliant challenge!

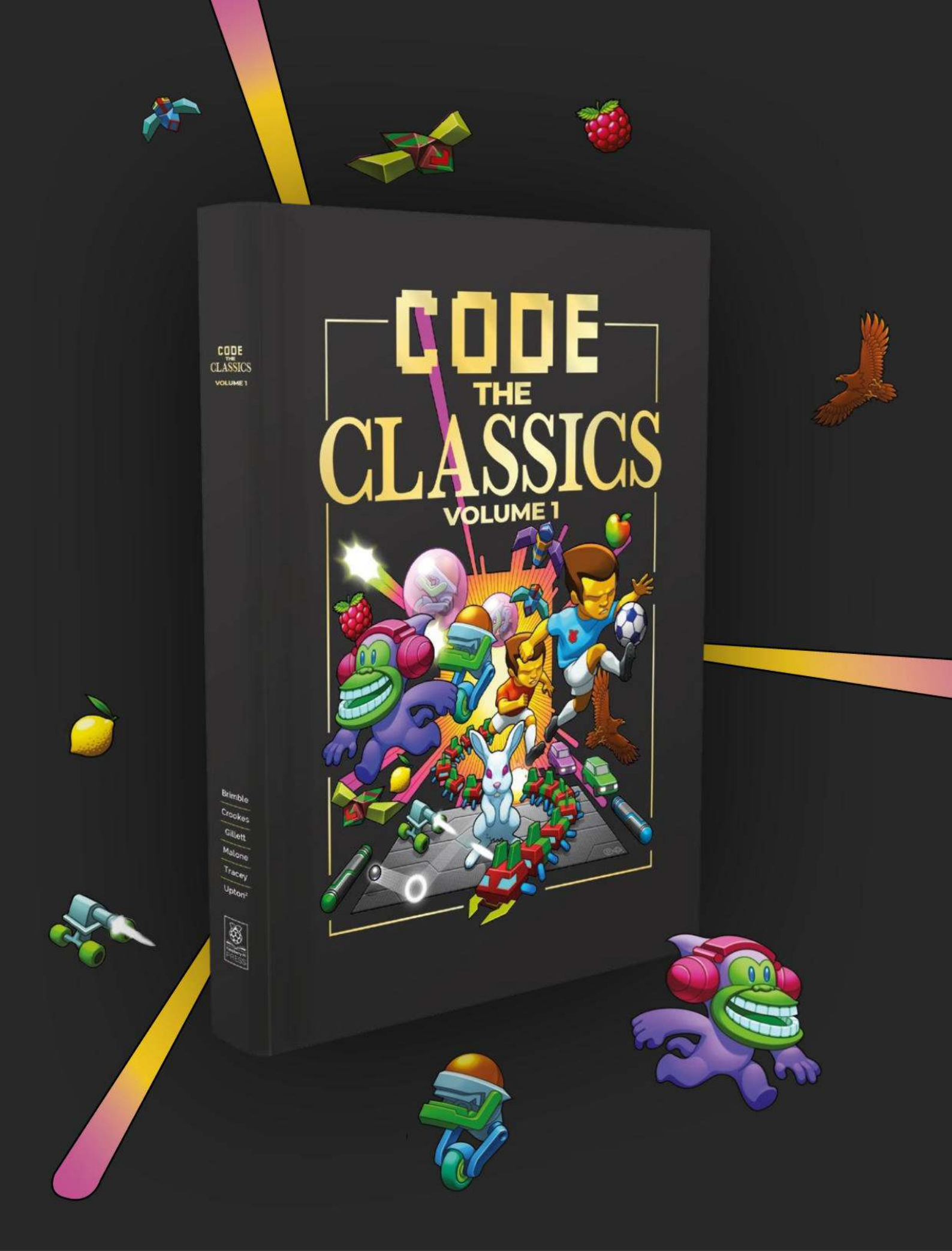
Figure 10
One of our many iterations of a twin-boom foam free flight plane

QUICK TIP

Put a very slight kink in the MIG wire so that the switch has some friction, helping to keep it open or closed.

Figure 11
The underside of the twin-boom plane design




CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²





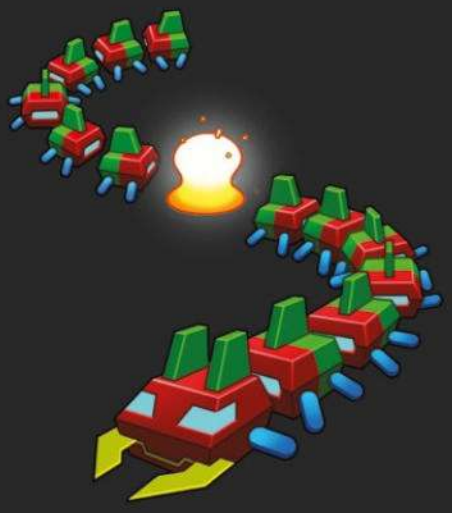
CODE THE CLASSICS VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

TUTORIAL



Metal stamping jewellery by hand

Learn how to use a humble hammer to create beautiful, one-off, and exclusive gifts. **Nicola King** gives this craft her stamp of approval



Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. Lately she's discovered the very relaxing aspects of hammering, stamping, and also prodding things with needles.

Hand stamping metal is a fantastic way to get creative, and to make bespoke, personalised gifts in a jiffy. In this accessible craft, in terms of both required skill level and the cost of entry, you can soon learn to impress your nearest and dearest (and even potential customers) with your metal stamping skills. With just a few basic and inexpensive tools, you can customise pieces with names, dates, and initials, as well as adding patterns and textures with specialist hammers if you so wish. Great results don't take long to achieve, even as a beginner. In this tutorial, we will take you through some of the basics around this popular branch of jewellery/gift making.

METAL STAMPING: THE NUTS AND BOLTS

In its most basic terms, metal stamping involves striking a piece of metal using a stamp to make an impression; for example, a letter, number, or other design, with a hammer providing the force. The hammers used tend to be fairly heavy in weight (ours is a 1lb hammer) with a wide, flat face to provide an even and correct angle so that the hammer blow is effective. The hammer that we've used has a brass head which helps to provide a 'soft' blow, meaning that there is less chance of unwanted movement, as this type of head has less reverberation than steel hammers, for example. As well as a hammer, stamping requires a metal

QUICK TIP

Make sure you are comfortable when stamping, as good ergonomics will result in better results and less stress on your limbs. Try standing up to stamp to see if it works better for you.

block to stamp onto, providing a support for the metal stamping process while also protecting your surfaces. Some blocks have central sections made of rubber which help to deaden the sound of repeated hammer blows.

If you'd like further information on the types of blanks that you can stamp onto, see the box overleaf for the particulars. We are going to use a variety of small blanks that we purchased online, and a selection of stamps accumulated in this author's jewellery-making stash over the course of several years. So, with our tools and hardware to hand, let's do some metal stamping.

STEP 1 IT'S ALL IN THE PREP

One thing that this author has learnt over the years is that it really helps if you have everything you need for your craft to hand, ready to go. There is nothing worse than realising that you don't have x, y, z, and then having to rummage around before you can actually get going. Working at the kitchen table, as we tend to do, there is plenty of room to assemble various stamping accoutrements, so make sure you're geared up with all you need.

Position your stamping block on a stable table so that you have plenty of room around it. Note that if your table is not stable, your stamp may 'jump' when hit, and your impression will be wonky. Our stamping block came with some adhesive 'feet' that meant it would not slip around on our surface. You can also add a piece of rubber underneath your block, as it helps deaden the sound of the stamping. It is worth noting that our block also had a film of oil on it (to prevent rust) which we just wiped away with a dry paper towel. Next, select your stamping blank and place it centrally on the block, and it's a good idea to hold it down with some tape. **Figure 1**, overleaf, shows how we marked the tape to show us the point above which we wanted to stamp. So, place the top edge of the tape where you want the bottom of your impression to be. For our first stamp experiment, we elected to make a name impression on a keyring blank. When it comes to the size of the blank to choose, do bear in mind that smaller blanks can be a little more fiddly to handle if you're new to the craft.

> **Bear in mind that smaller blanks can be a little** more fiddly to handle if you're new to the craft

QUICK TIP

If making a jewellery item for an individual, be aware of any allergies they may have to various metals. It's always a good idea to try and plump for blanks that are lead and nickel free, for example.



Above
A selection of the tools and consumables required for metal stamping. Some kind of sticky tape, or washi tape, is extremely useful in order to hold the blank in place on the block while you stamp

STEP 2 COMMITTING METAL TO METAL

Next, select the stamp that you wish to use, whether it be lower- or upper-case letters, numbers, or perhaps a design such as a tree, leaf, symbol, and so on. There are many different font styles available in letter/number stamps. We selected our font and picked out the letters required for the name.

Position and hold your first stamp on the blank (we used our left hand as we're right-handed) and gently pull the stamp towards the tape guide mark that you made, until you feel the stamp catch the edge of the guide. With your stamp vertically straight and anchored in position, use the hammer in your dominant hand to give the stamp one or two, at →

YOU'LL NEED

- **A stamping hammer** (you can try using just a regular hammer)
- **A small metal stamping block**
- **Metal blanks**
- **A stamp set** (letters, numbers, or both)
- **Design stamps** (optional)
- **Chain nose pliers**
- **Washi tape/ guide tape** (optional)
- **Enamel marker pen** (highly recommended for definition)
- **Metal punch** (optional)
- **Tape measure/ ruler**
- **Jump rings, keyrings, and chains** (depending on what you are making)
- **Dry cloth/paper towel**

TUTORIAL



Figure 1
Use a ruler to mark out on the tape where you want the impressions to be on your blank. You will be stamping above the mark you've made

QUICK TIP

Ensure that your hammer, metal block, and stamps are all covered and stored carefully when not in use in order to protect them from rusting. This author left her original stamping block in a cold conservatory, and lived to regret it.

Figure 2
Believe us, it will look better once we've cleaned it up!



A METAL STAMPING STASH: TOOLS AND KNOWLEDGE

- The internet is always your friend when you are looking for the trappings of, and guidance for, a new hobby, and there are many purveyors of jewellery-making goodness who sell stamping paraphernalia under that 'metal stamping' umbrella. **Cooksongold.com** is a jewellery-making supplies site that has pretty much everything you can think of when it comes to this craft. Always compare prices, but we found some reasonably priced stamping blanks and stamps in a recent sale.
- Kits just make it all so easy! Of course, if there was ever a gap in this market, it was quickly plugged and there are numerous kits available which will provide you with all that you need to get started. As always, Amazon is a good place to peruse the various offerings: **hsmag.cc/StampingKit**.
- There are plenty of helpful videos online that offer viewers a good understanding of metal stamping, what you'll need, how to do it, even how to sell your professional-looking makes, and so on. A very useful YouTube channel is ImpressArt Stamps (**hsmag.cc/ImpressArt**) and we've used a few of this company's products in our tutorial.
- There are also, naturally, some books on the subject that would be worth a browse. ***DIY Stamped Metal Jewelry***, by Adrianne Surian, is one such volume.

the most, medium to hard hits. This should create a good, strong impression. It's much better to give the stamp one or two hits than lots of little taps, which are unlikely to be as effective. If you hit it multiple times, you also risk moving the stamp; if that happens, it can result in a blurred image, which is not the effect we want.

Remember that you'll get more familiar with the degree of force needed to strike a particular stamp when you practise. It can be difficult to know how much pressure/striking force to use initially, so if you have some misaligned preliminary attempts, don't worry about it – practice makes perfect! You can try stamping on a piece of chipboard/wood if you have some to hand – that would save your blanks, and would give you practice in holding the hammer and striking down on the stamps.

STEP 3 THE FINISHING TOUCHES

Once you have stamped your design or name, remove the tape that you used as a guide. Hopefully you are happy with the indentations that you have made, but you might feel that they don't stand out enough. The one thing this author would highly recommend purchasing is an enamel marker pen. If you use one of these, it will make your impressions much more obvious, and you can purchase them in a range of colours (black, gold, silver, brown, green). We chose black, and are really happy with the results.



Above
The head of this hammer is made from brass which absorbs some of the shock and also helps prevent damage to stamps



To use, first, shake the pen a few times to get the solution moving. When the enamel has flowed to the tip of the pen, draw over the impressions you've made, and make sure to try and get into the nooks and crannies of the piece. **Figure 2** shows one of our pieces where we used the pen. Leave the enamel to set for around three or four minutes. Then, with a dry paper towel/cloth, very gently wipe away the excess. Be careful not to pull the enamel out of the indents you've made. It's that simple! Hopefully, you can really now see the stamped impression much better.

If you want to polish your piece of stamped metal, now is a good time. Add any jump rings, chains, keyrings, or whatever you need to turn your blank into a piece of jewellery or a useful item, and hey presto, one gift ready to be given!

An attainable craft, with minimal financial outlay, if you enjoy making quick gifts, we highly recommend giving hand metal stamping a try. The hammering can be a little on the noisy side, but many makers are probably used to that anyway in some form or another, and it's definitely worth it!

QUICK TIP

Possibly something to add to your stamping arsenal at some point is a 'strike jig' which holds stamps in place horizontally, and may help protect fingers.

Above
A selection of letter stamps – there are many stamp options in various fonts to choose from

A LOWDOWN ON STAMPING BLANKS

There is a very good choice available in terms of what you can actually stamp onto. Firstly, let's think about the shapes available. You can choose from dog tags, cuff bangles, cuff rings, ID plates, washer shapes, leaf shapes, teardrop shapes, hearts, pendants, and so on. This author has even seen some blanks in the shapes of cats, dogs, and squirrels – the point being, the choice is wide. Some of these blanks will have a small hanging hole in them already, but if not, you can use a metal punch to make a hole in order to hang the item as a charm on a necklace etc.

Secondly, let's think about base metals. Assuming that you are probably new to the craft, stamping onto the more expensive base of sterling silver may be a little over-ambitious until you have honed your skills and are more confident in your work. Silver is the premium quality choice for stamping as it's a malleable metal that holds impressions well, but it's expensive. Some of the most popular, and far more pocket-friendly, options include:

- **Aluminium** – strong but workable, and can be polished to a high shine
- **Pewter** – heavier than aluminium, but still soft to stamp
- **Copper** – soft, inexpensive, easy to find, but prone to tarnishing
- **Stainless steel** – a harder metal, so more challenging to stamp on for a beginner
- **Brass** – a weightier metal that is popular for stamping and engraving

Whatever you choose, you are sure to get pleasing results… after a little practice!

QUICK TIP

Bear in mind, the more complex the stamp, the harder you may have to strike it. For example, if using letter stamps, the letter 'L' is less complex than the letter 'W'.

MicroPython Bluetooth serial

Send data to Pico W without needing wires



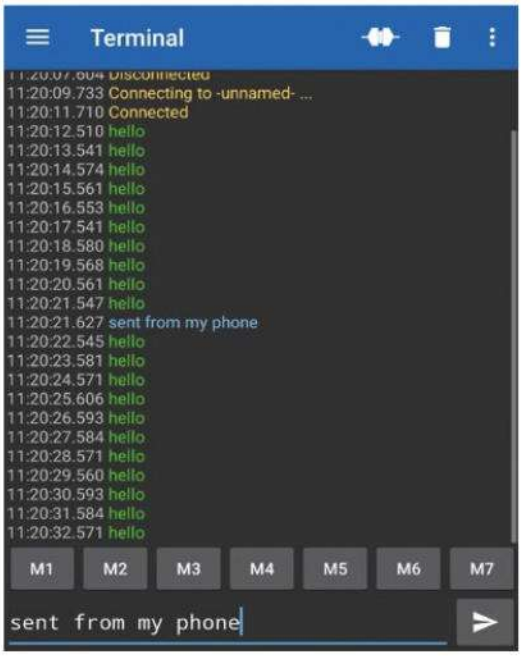
Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

The humble serial terminal is one of the easiest ways of finding out what's going on with your microcontroller. Almost all languages have something like a `print` function that will send text to your computer, and there's usually something like `receive` that takes text in the opposite direction. This is all great while you can wiggle a cable through the enclosure and plug it into a suitable port on your controller, but what happens when you've fully enclosed your project? What if it's buried deep inside the guts of your latest invention?

Thankfully, this method isn't limited to USB connections. It's not even limited to two-wire UART connections. In this article, we'll look at sending and receiving text over Bluetooth using Raspberry Pi Pico W.



Right
You can pretend that you're an important, busy person with lots of messages to reply to

Bluetooth is an incredibly versatile protocol. In fact, it's not even really one protocol – there's Bluetooth classic and Bluetooth Low Energy (BLE), and we'll be using the latter. On top of this underlying protocol, there are ways of sending almost any form of data. Unfortunately, this flexibility brings in a large amount of complexity. Raw Bluetooth can be a bit of an unwieldy beast to work with.

Since getting data on or off an embedded Pico W is a common problem for us, we've decided to build a MicroPython module that lets us send and receive data over a BLE UART in much the same way we'd use a USB UART.

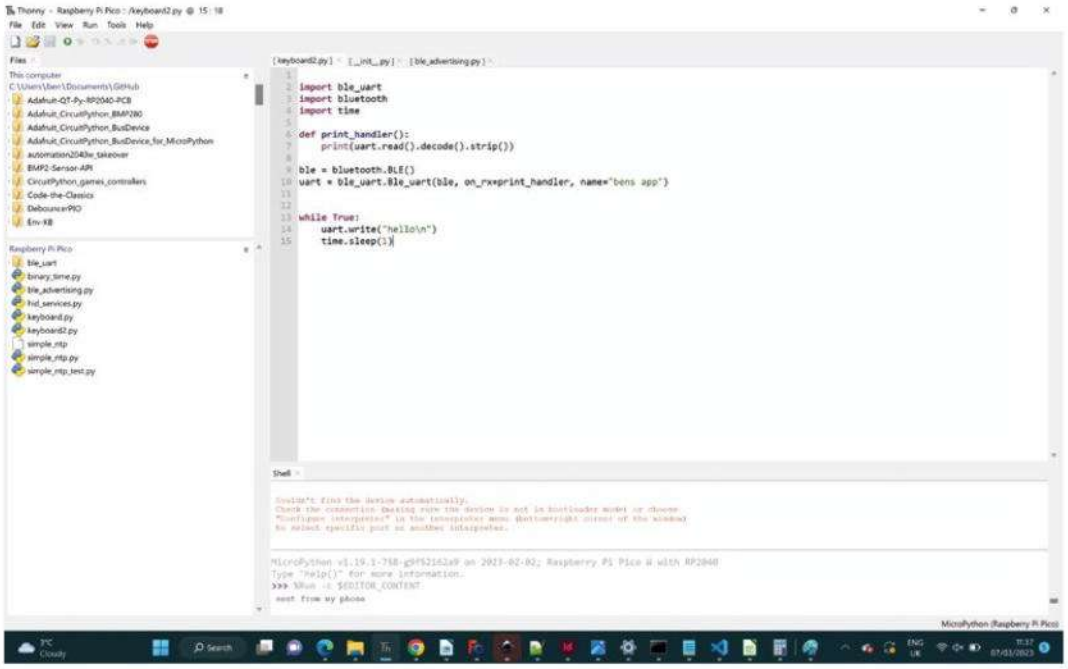
First, you'll need a Pico W with MicroPython flashed to it, and Thonny installed on your computer. If this is the first time you've used MicroPython on Pico W, take a look at **hsmag.cc/mpgettingstarted**. Bluetooth support is only in the latest version of MicroPython for Pico W, so you'll need to grab that if you haven't already.

We've created this tutorial using a pre-release version of MicroPython in the hope that it's available by the time you read this. However, it's possible that it has been delayed. With the latest MicroPython flashed to your Pico W, type the following in the Thonny Shell:

```
import bluetooth
```

If this throws an error, it means the version you have doesn't support Bluetooth. If it runs it normally (and outputs `>>>` on the next line), then the version you have does support Bluetooth and you should be good to go.

Once you've got a supported build of MicroPython, the next thing you need is the ble_uart module. You can download this from here: **hsmag.cc/BLE_UART**. You'll need to copy everything over to Pico W. Unfortunately, there's no drag-and-drop method for doing this, so you'll



Left
Once everything is set up, just call 'write' to send messages into the ether

need to open each file individually in Thonny and use 'Save as' to give it its same name on Pico W (you'll also need to create the folder).

Once it's all there, you can open **test.py** and run it. It should start without errors, but it might not be obvious that it's doing anything, and that's because Pico W is only half of the communication. You also need something to receive the data.

There are lots of ways of receiving serial over Bluetooth. We've been testing the Android App 'Serial Bluetooth Terminal' by Kai Morich, but there are others that should also work.

In this app, you'll first need to connect to your running code on Pico W. You can do this by opening the menu in the top left-hand corner and selecting Devices. Make sure you're on Bluetooth LE and select Scan, and your device should come up. Tap it and you should connect – you'll start seeing Hello appear every second. Let's take a look at the code we're running now to see what's happening.

```
import ble_uart
import bluetooth
import time

def print_handler():
    print(uart.read().decode().strip())

ble = bluetooth.BLE()

uart = ble_uart.Ble_uart(ble, on_rx=print_
handler, name="bens app")
```

> It's a useful one for helping keep an eye on our projects by **sending data back and forth without needing to resort to stretched lumps of metal**

```
while True:
    uart.write("hello\n")
    time.sleep(1)
```

As you can see, once everything's set up you can just call the `write` method of the `ble_uart` object. If there's a device connected, this will send data there; if there's not, it won't do anything, so you can use this to send debugging information and just connect to the device if you need to. Remember though, that there's no security on this, so don't send sensitive information.

We can also send information the other way. If you look at how we set up the `uart` object, you'll see we pass an `on_rx` parameter. This is a method to call when data is received over the UART. In this case, we just convert it into text and write it to the main terminal. However, you could also use this to extract some commands or data from the text, and implement these in any way you like.

Bluetooth can do many things, and we have looked at just one, but it's a useful one for helping keep an eye on our projects by sending data back and forth without needing to resort to stretched lumps of metal.

STOLEN CODE

For the most part, this isn't our code. We've borrowed it from the Bluetooth examples in the main MicroPython repository. However, we've wrapped it up as a module to make it easy to drop into any Pico W projects.

TUTORIAL

Slicing for speed

How to get quick prints with PrusaSlicer



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Above
Organic supports have a certain 'monster from the depths' vibe when printing boats

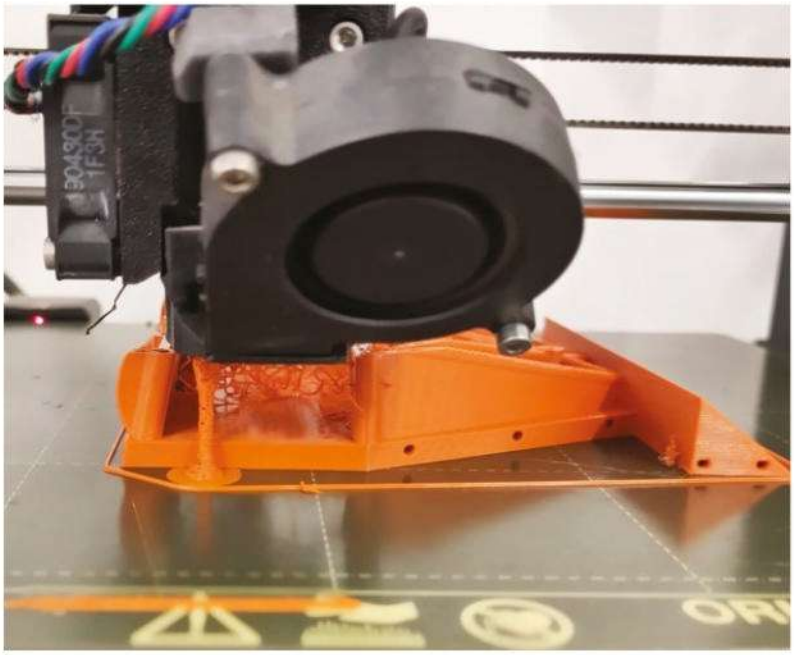
3D printing is part of a set of technologies that are often called 'rapid prototyping'. This name can often feel ironic when you set off a ten-plus-hour print.

However, with a bit of tweaking, you can really start to bring those print times down, especially if you can compromise on strength or aesthetics – as you often can when prototyping something.

In this article, we're not going to look at tuning the printer – that's very hardware-specific and, while it can have significant speed improvements, it's a deep rabbit hole that can lead to many failed prints along the way. We'll be looking at pure slicing improvements that should speed up a print on any hardware.

The first thing most people learn when they start 3D printing is that layer height is a compromise between speed and looks. It's a pretty simple trade-off. The only real problem with chunky layers is it obviously limits how accurate your model can be. If you have features that you need to include that are very small, chunky layers simply can't reproduce them.

Almost everything else comes down to 'use less plastic'. This is good for speed, good for your costs, and good for the planet. Let's look at the ways we have of doing this.



Above
Our testing has shown organic supports can be a bit prone to 'spaghettification' when not on the print bed

> **One feature of PrusaSlicer 2.6 that can help here is** the new tool for slicing a print into parts

SUPPORTS

These are much maligned in some 3D printing circles, but they're a necessary part of printing. With modern slicers, they are not usually too hard to remove. If you can print without supports, it's almost always quicker to print without them. The first and most obvious trick is to orientate your parts so that they don't need supports. Sometimes this is possible, but sometimes it's not.

One feature of PrusaSlicer 2.6 that can help here is the new tool for slicing a print into parts. There's nothing revolutionary about this, but now you can add pins and holes so that they push together neatly and in exactly the right position. This can make the glue-up easier and more accurate. This obviously involves an extra step after the print, but for some parts (i.e. if it allows you to avoid a significant amount of support material), it could be a good trade-off.

You can cut an object by highlighting it and selecting the Cut tool in the left-hand tool panel. You can move or rotate the cut plane to select where you want to cut. Click 'Add Connectors' to add pegs or dowel joints. They don't add automatically, so you need to select where you want them to go (you only need to place them on one part – the other part will be added automatically). Once you've got everything set up, you can press 'Perform Cut' to split your part.

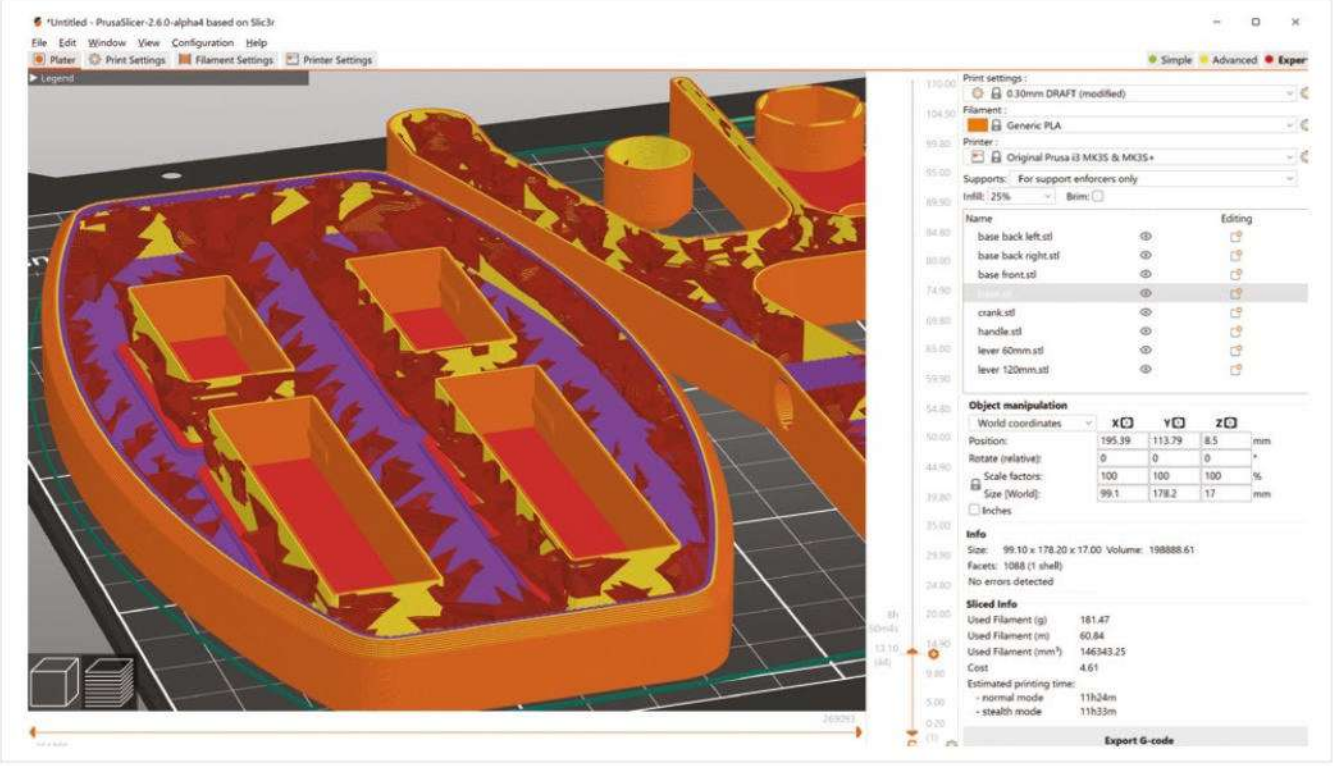
If you do need supports, it's useful to minimise the number of them used. If you just select 'Supports: Everywhere', then the chances are you'll end up with far more than you need. You can use support enforcers to just get the supports where you need them. There are a few ways of adding support enforcers. The 'traditional' way is to add a shaped enforcer. Right-click on an object, and then select →

VERSION 2.6

Some of the features we've looked at in this article are only available in PrusaSlicer 2.6. At the time of writing, this is still in a pre-release state. By the time you read this, the final version might be available from **hsmag.cc/PrusaFinal**.

If it's not, you can get the pre-release version from the GitHub page: **hsmag.cc/PrusaSlicerGitHub**.

It's possible that there will be some small changes between the version we're using for this tutorial and the final release version.





Above
You can see the Lightning infill coming out from the sides to support the top

Right
Organic supports

'Add support enforcer'. You can then add a shape, and anything inside this shape will get supports (if you enable 'Supports: for support enforcers' only). In more recent versions of PrusaSlicer, you've been able to paint on where you want the supports to go. From the left-hand toolbox, select the paintbrush icon, and you can draw on the areas where you want supports (again, you have to enable supports for enforcers).

In the very latest version (PrusaSlicer 2.6), it will automatically apply this paint where it thinks you should add supports. Click on the paintbrush, then 'Automatic painting'. This is far better at working out where you need supports than if you just enable 'Supports: Everywhere'. It is pretty confusing that there are two ways of automatically adding supports that have vastly different results and are accessed in completely different ways (wouldn't a 'Supports: Minimal' or similar option make far more intuitive sense?). Alas, this is the interface we have for now.

We've been using automatically painted supports for a while now, and have been really impressed with the results. They're great at saving material and haven't let us down yet.

Once you've decided where you want supports, you need to decide what type of supports you want.



Above
This Lightning infill will branch out to support the top of this print

There are (as of PrusaSlicer 2.6) three options: Grid, Snug, and Organic. Grid are the default supports. Snug are – as the name suggests – much smaller. Snug supports are particularly good if you've got a tall object that only needs supporting quite high up. For example, if you've got a box with an unusually shaped cutout in a wall. Grid supports would probably support this from the build plate, while Snug supports would just support it from the bottom of the cutout. There can be significant time-saving by switching between these two.

Organic supports are new, and they are tree-like structures where a single trunk can branch out and support a wide canopy. For some shapes, particularly those that need support across a wide area, they can be a great option. We have had great success using Organic supports when they're on the print bed, but quite a few problems when they're on top of part of the print. We'd suggest a bit of caution there.

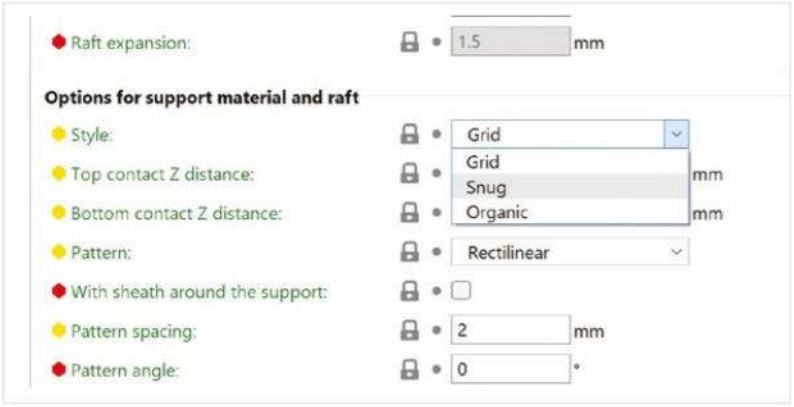
We're recommending slicing with both Snug and Organic supports, and seeing how it affects print times – it can be quite dramatic in either direction.

Grid supports – though we've used them quite a bit in the past – aren't usually a good option for fast printing. You can select the support type in Print Settings > Support Material > Style.

STRENGTH

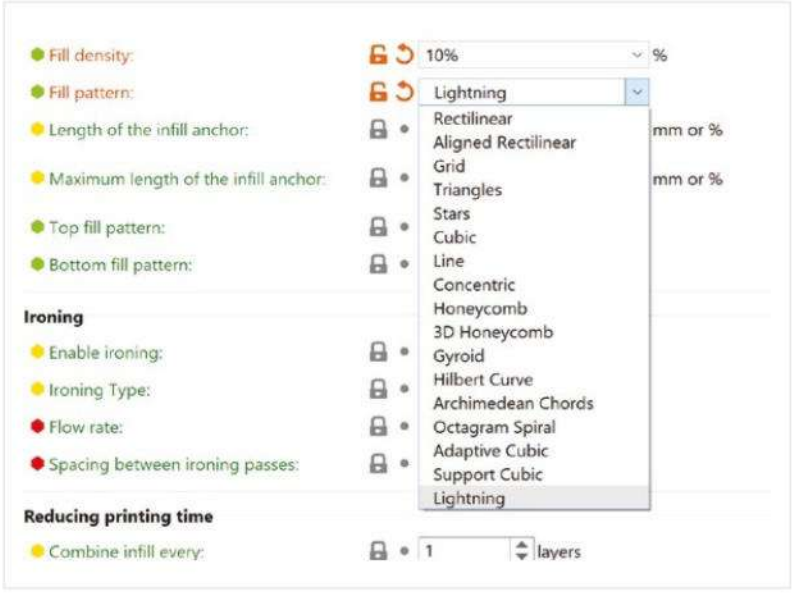
We've looked a lot at speed, but that's not the only consideration. What about strength? Particularly with the Lightning infill pattern, you can end up with weak parts. The fastest way to increase the strength of a plastic part is to increase the thickness of the outer wall. You can do this by increasing the number of perimeters, but this adds to the print time. Alternatively, you can make each line of filament thicker – this is known as the extrusion width. You can increase the perimeter extrusion width so that you can get the same amount of thickness from fewer perimeters. There are limits to how far you can go with this, and we'd recommend not going further than twice the nozzle diameter. You can change this under Print Settings > Advanced > Extrusion Width > Perimeters.

The slicer will adjust this slightly to fit the features of the model, but it shouldn't have much of an impact – particularly on side walls, which tend to be the places where strength is important.

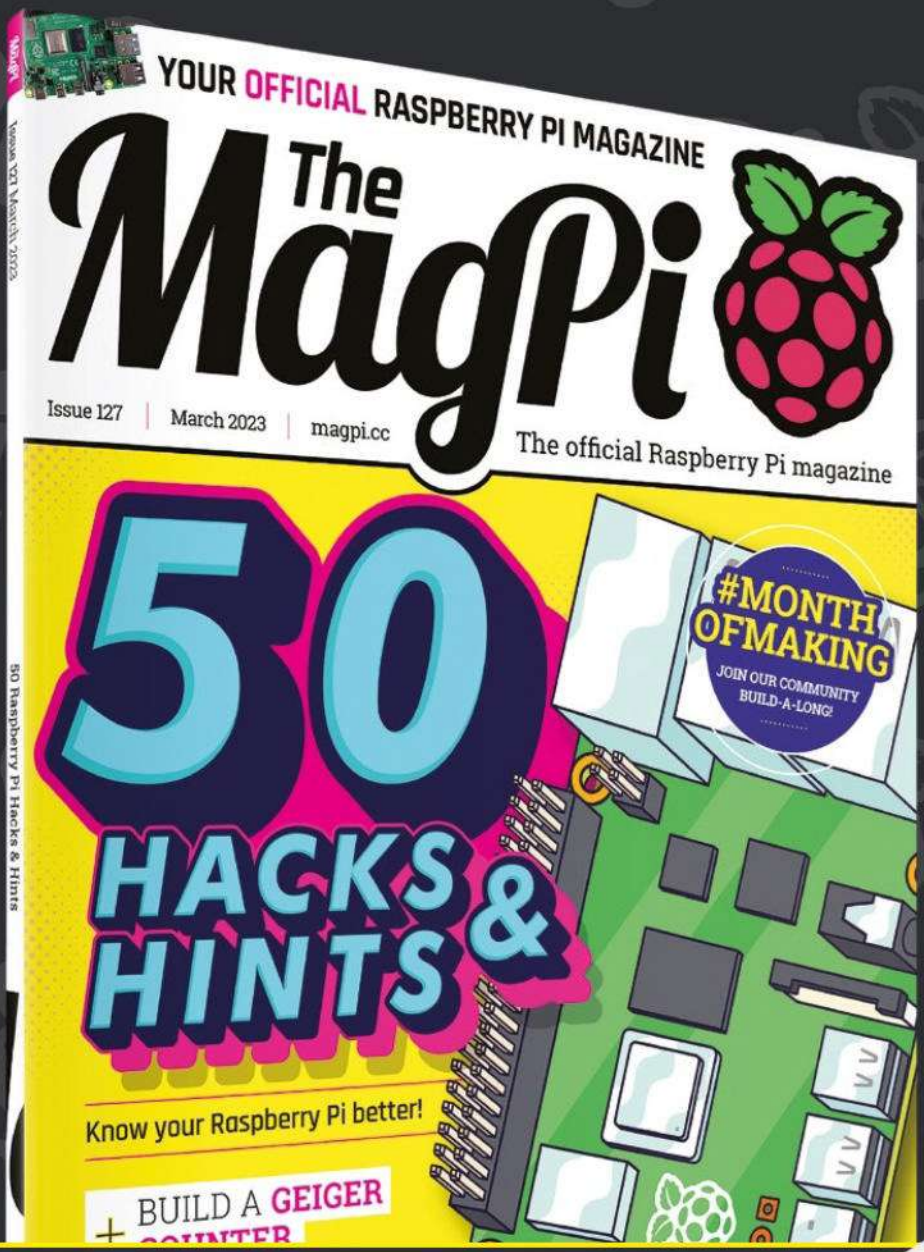


Above
Organic supports are getting a lot of press at the moment, but our experience has been that Snug supports are often faster, especially with the automatic support painter

Below
There's a lot of infill types to explore



DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE FOR JUST £10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door



+ FREE RASPBERRY PI PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.

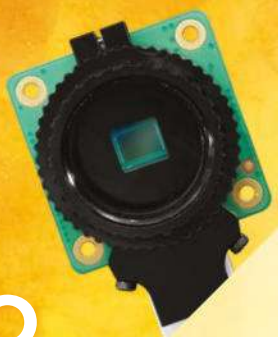
magpi.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated



PG 90



SCORPIO

For when you need all the LEDs

PG 92



SPINTRONICS

Learn electronics without electricity

PG 94



DEBUG PROBE

Microcontrollers made easier

PG 84



BEST OF BREED

Robotic pets on show

PG 96

CROWDFUNDING

Sift electrons from the wind

ONLY THE BEST

Meet your new robotic best friend

Are robo-dogs or mechanical cats in your future?

By **Marc de Vinck** @devinck

Will your next robot be a little less humanoid and look a little more like your best pet friend? I think most people imagine robots in a human-like form, or even just as a mechatronic arm, slinging parts around a factory floor, but there are other forms that may not be as functional, but certainly more adorable. Enter the dog-bots! And cat-bots too!

Over the last few years, small dog- and cat-like, robots have shown up everywhere online. There used to be only a few, but after Boston Dynamics and its masterful marketing team launched the dog-like robot called Spot, it seems like a whole army of similar-looking robots have hit the market. But, unlike that famous dog-bot, which costs upwards of six figures, most of the other models on the market are far less expensive, albeit far less sophisticated too.

Many of the animal-like bots on the market today are clearly just for fun, as they are very limited in their capabilities, but there are some that are quite competent in carrying small payloads, or surveying an area outdoors. None seem to be as cute and cuddly as the real thing, but you also won't need to carry a doggy bag or change the little box with this collection of bots. So, let's take a look at some robo-dogs and cats that are a bit more than just toys, but also aren't going to cost more than a car.



PuppyPi Pro vs Mini Pupper

HIWONDER $920 | hsmag.cc/PuppyPi

MINI PUPPER $559 | hsmag.cc/Minipup



The PuppyPi Pro by Hiwonder is a great example of a dog-like robot that has a lot of functionality. This quadruped is compatible with the Robot Operating System (ROS), and has a lot of advanced features. The bot is powered by a Raspberry Pi, so you know you can add an almost unlimited variety of sensors and accessories. It comes equipped with lidar and an AI vision system that will allow for simultaneous localisation and mapping, aka SLAM, which is an advanced capability for our robot under $1000.

The PuppyPi is available in three different versions, ranging from $700 to just over $900. If you have the budget, I'd recommend grabbing the Pro version, as it includes the lidar module which you will certainly want after you get familiar with the platform.



The Mini Pupper is another ROS-compatible robot, which already makes it a good choice for anyone exploring the world of robotics. The Pupper is powered by a Raspberry Pi 4B 2GB, which allows you to add a variety of Raspberry Pi-compatible sensors and accessories. It also runs OpenCV for vision and handles all the communication for movement.

The creators have a decent guide to programming the bot, including machine learning and SLAM examples. And, since it's open-source, the creators also provide mechanical design files and the PCB layouts so you can customise it however you'd like.

VERDICT

PuppyPi Pro

Raspberry Pi-powered!

10/10

Mini Pupper

A more affordable ROS-compatible robot companion.

9/10

BEST OF BREED

RASPBERRY PI HIGH QUALITY (HQ) CAMERA – 12MP

RASPBERRY PI ◆ **$50**

hsmag.cc/HighQualityCam

One common thread in most robotic quadrupeds is the ability to see. And the people over at Raspberry Pi just upped the game with the new Raspberry Pi High Quality (HQ) Camera. It features a high-resolution 12.3-megapixel sensor with added low-light performance. Just remember, when buying this module, that you will also need to pick up a lens, which is not included.



Unitree Go1 Air

UNITREE ◆ **$2700 (+$1000 shipping)** | **hsmag.cc/Go1Air**

> "It's still a very good value when you look at the specs and advanced capabilities"

The Unitree Go1 ups the game a bit when it comes to features and technology. It also comes at a higher price, and considerable shipping cost. And, although it is more expensive than other bots in this roundup, I will also say it's still very good value when you look at the specs and advanced capabilities.

The Unitree bot has a suite of stereo-depth cameras, foot force sensors, a microphone, and speaker. It also features a 16-core CPU + GPU, and advanced motors in the joints for a high level of control, and on-the-fly adaptability with its compliant joints. You can use a lapel controller to have the bot follow you at your side, not behind you like most others do, which is a nice feature. It's also controllable via a computer, or even a remote control.



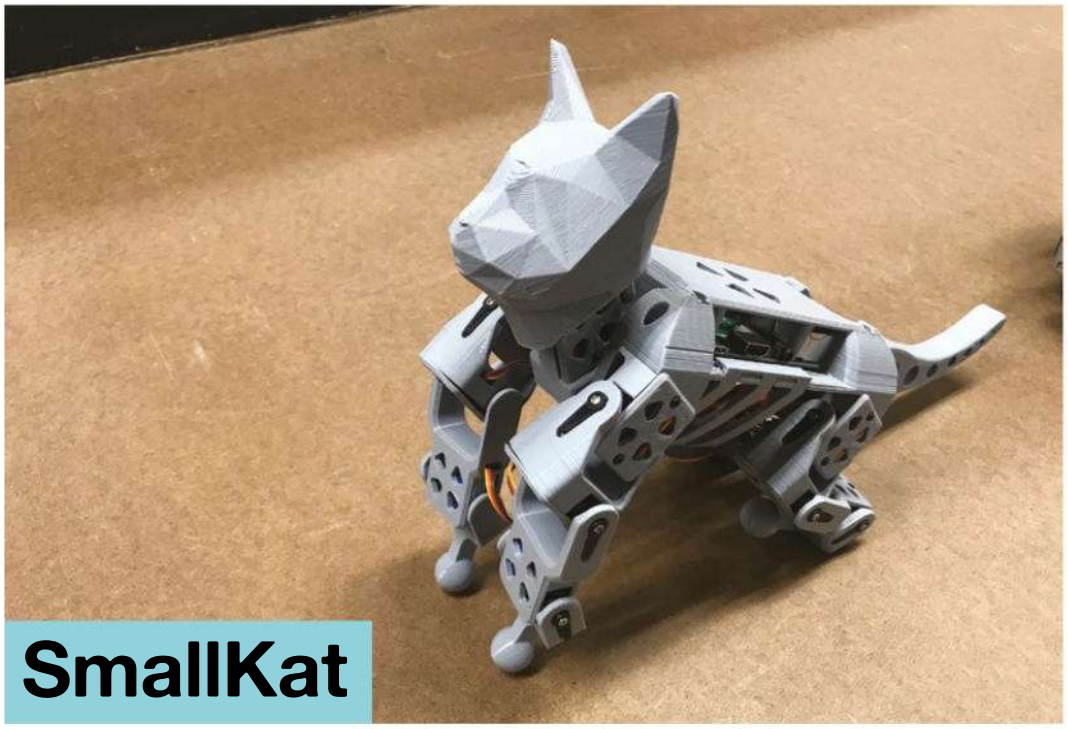
Left ◆
A pet with more processing power than our laptop

VERDICT

Unitree Go1 Air

Expensive, but still good value.

10/10



SmallKat

TINDIE $500 | hsmag.cc/Kat

Left
Cute 3D-printed cat, or horrifying vision of a dystopian future?

Not all bots need to be dogs! The SmallKat is a 16-DOF robot that is open-source, includes mostly 3D-printed parts, and features a full dynamics controller. Made by Kevin Harrington of CommonWealth Robotics, the SmallKat is a great introduction to the world of robotics. It's a complete kit, including all the 3D-printed parts, 16 micro servos, a Wii Classic controller, breakout boards, battery, and ESP32 dev board. The SmallKat has an IMU with integrated dynamics controller, perturbance compensation, and BowlerStudio models for simulating your code. It makes for a great educational kit, as the creator has a series of recorded videos, lesson plans, and source code, all available for free online.

VERDICT

SmallKat

Open-source with full dynamics controller.

9/10

16 MM 10MP TELEPHOTO LENS

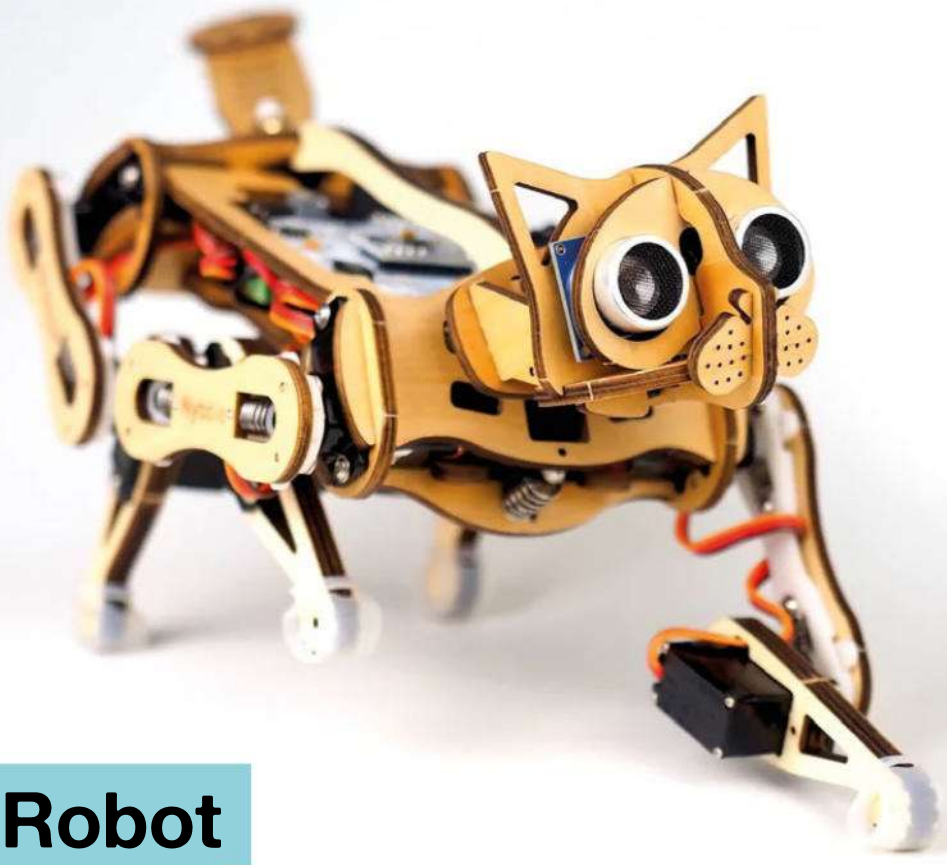
ADAFRUIT $50

hsmag.cc/TelePhoto

If you get the new Raspberry Pi HQ Camera, then you'll need a lens, and the 16 mm 10MP Telephoto Lens might be perfect. This lens is good at capturing objects at a distance. It includes a thumbscrew for easily adding it to the HQ Camera. If you need to be able to identify objects at a distance with your Raspberry Pi HQ Camera, then look at this lens.



BEST OF BREED



Right
It's not aggressive, it's just got half a bite

Petoi Robot Cat Nybble

PETOI $269 | hsmag.cc/Nybble

Yes, another cat has entered the arena! Meow! This time its basic structure is laser-cut wood, as opposed to 3D-printed. The Petoi Robot Cat Nybble is designed to be easily assembled thanks to its screw-free interlocking puzzle-like frame. You can program the robot via C++ or Python, and there is even an app for iOS or Android. And, since it's open-source, you can modify or add sensors as needed. If you want to add even more power and possibilities, you can easily add a Raspberry Pi to the platform and perform more advanced tasks.

> "You can easily add a **Raspberry Pi to the** platform and perform more advanced tasks"

VERDICT

Petoi Robot Cat Nybble

A classic laser-cut kit? Kitty!

9/10

6MM 3MP WIDE ANGLE LENS FOR RASPBERRY PI HQ CAMERA – 3MP

ADAFRUIT $25

hsmag.cc/WideAngle

If you want a broader field of view for your project, then the 6mm 3MP Wide Angle Lens from Adafruit might be a perfect match for your Raspberry Pi HQ Camera. This lens has a wider view angle compared to its telephoto counterpart, which gives you a more CCTV-like view of its surroundings. A good choice for projects like a video doorbell or security camera.



Free eBook!



Download your copy from

hsmag.cc/freecadbook

REVIEW

Feather RP2040 Scorpio

Do your projects need more LEDs?

ADAFRUIT $14.50 | hsmag.cc/Scorpio

By Ben Everard @ben_everard

You may well have used WS2812B LEDs (aka NeoPixels) in your projects before. They can be really easy to use, as they just need three connections – power, ground, and signal – to drive lots of LEDs. You can put ten together to drive a bar-graph output, or even a matrix of 64. They're generally well-supported by almost all hobbyist microcontroller libraries.

However, while in theory you can drive an unlimited number from your microcontroller, in practice, the more you try to connect, the more problems you're going to have. The Scorpio is designed to solve a few of these issues:

- The 3.3V output on most modern microcontrollers is below the minimum voltage for WS2812B LEDs. In most cases, you'll get away with it, but sometimes you won't. If you've got slightly longer cables or just an LED with tighter tolerances, it'll stop working. The Scorpio boosts the output of the eight LED pins to 5V to avoid this problem (there's a cuttable selector to bring these back to 3.3V if you need to).

- You can chain an arbitrary number of LEDs together, but each LED you add slows down the updates. For small numbers, this isn't a problem, but if you've got longer chains (particularly if you're trying to use the HDR function – see below), this becomes a problem. The Scorpio has eight-channel output, so you can split your LEDs up into eight sections and send the data out in parallel to minimise this problem.

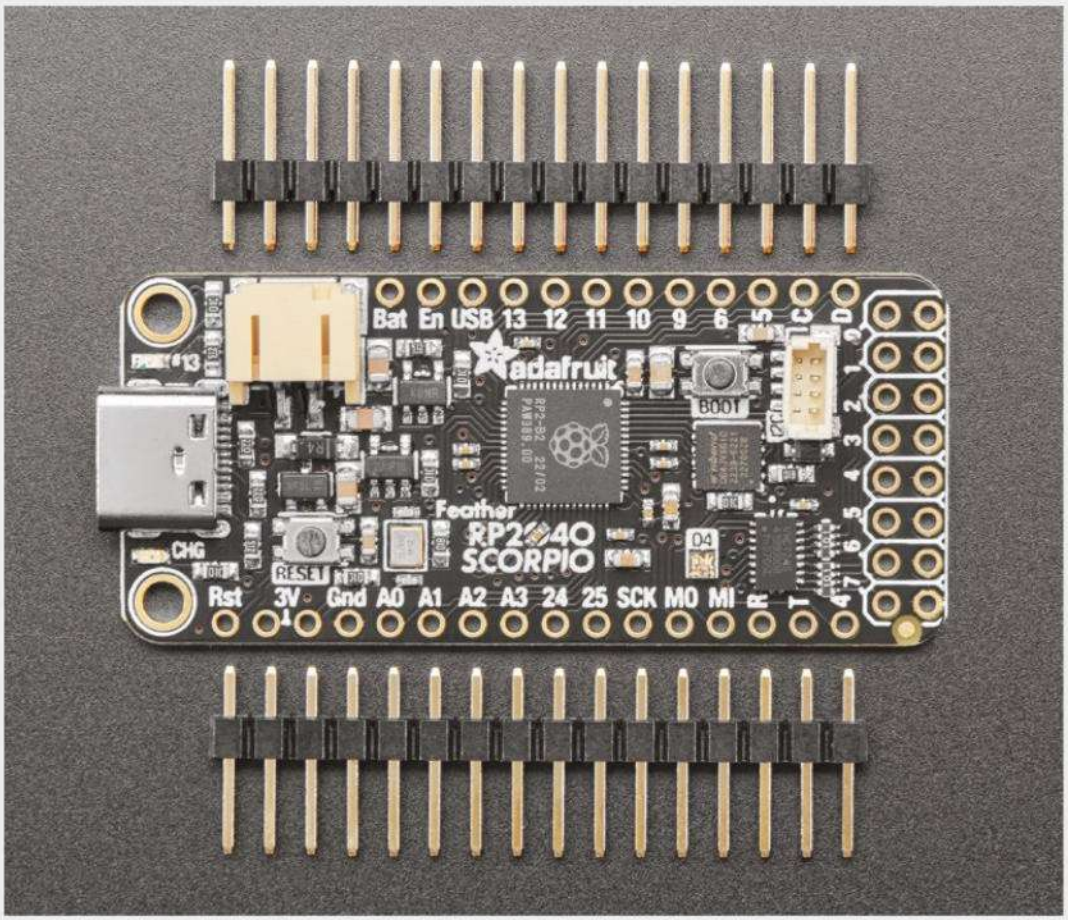
- Each LED output also has a ground connection, which makes wiring simpler (there's no V+ because, typically, you want to power them from an external supply if you're using enough to make this board useful).

Below
Debug, boot, and USB connections are broken out on the bottom of the board



In addition to the LEDs, there's a lot of GPIO pins for other hardware. There are 21 3V GPIOs (in addition to the eight 5V LED GPIOs), two of which are also on a STEMMA QT / Qwiic connector that makes it easy to add on peripherals.

One slight challenge with RP2040 boards is the boot button – if you want to change the firmware, you need to hold this down at reset to put the board in bootloader mode. This can be an issue if the board is buried deeply inside a project. There is a solution where you can enter the bootloader by double-tapping reset, but this requires firmware support. The Scorpio breaks out the boot button on a surface-mount pad on the bottom of the board, so you can add your own more accessible button if you need to.



Left
There's a lot of hardware crammed onto this little board

> The Scorpio breaks out the boot button on a surface-**mount pad on the bottom of the board**

The D+ and D- lines from USB are also exposed, so you can solder on your own USB connection as well if needed.

Hardware is only any good if there's software to help you use it. Again, this is an area where this board shines. The NeoPXL8 library brings support for eight outputs to both Arduino and CircuitPython, and the HDR (Arduino-only) sub-class uses dithering and lets you display 16-bit colours (rather than the standard 8-bit).

This software is easy to use and well-documented. If you're familiar with either Arduino or CircuitPython, you should find it easy to get started.

The elephant in the room on this device is connectivity. There's no Wi-Fi or Bluetooth on board, so you can't control it remotely without additional hardware. This is one of those things that you either need or you don't. If you need Wi-Fi, then you'll probably want a different solution. If you don't, then it's not a problem.

There's nothing on the Scorpio that you can't add to other boards. Level shifters are available, and most of the other features are really there to help you simplify the wiring. That said, simple wiring is a really nice feature when you're trying to make your project maintainable and understandable, and even better if it needs to be small. Scorpio isn't going to revolutionise the field of LEDs, but it is going to make life a little bit easier for people who like adding a lot of LEDs to things. It'll reduce the temptation to try and run LEDs with low voltages (which, in turn, reduces the headache of figuring out why something that worked as a prototype has suddenly stopped working when you build it out), and it'll keep projects neat and tidy.

VERDICT

One more reason to add more LEDs to everything.

9/10

REVIEW

Spintronics – Build mechanical circuits

Learn electronics by making moving mechanical contraptions

SPINTRONICS ◆ **£80 inc. VAT** | **upperstory.com**

By Rob Miles @robmiles

One of the difficulties in understanding electronics is that you can't see electricity as it goes about its business. A flat battery looks exactly like one which is fully charged, and a wire connected to 1000 volts looks just the same as one connected to nothing. It is only when the current flow makes a light turn on, or a motor spin, that we can see electricity at work. Spintronics aim to fix this by creating physical mechanisms that behave in the same way as electrical circuits. They have created mechanical versions of switches, resistors, capacitors, coils, and even transistors for learning electronics. These can be used to build an impressive range of devices, including logic gates, voltage dividers and, if you purchase the add-on to the starter kit, you can create oscillators, flip-flops, and all kinds of other devices too. It is also great fun just playing around with the components. They are just lovely to tinker with.



Right ◆
The sets are packaged as 'acts' in a story which is wrapped around the exercises. These are the components for Act 1. At the top left are the metal baseplates you build your mechanisms on. Plates can be locked together to create larger areas. In the middle is the large pullstring-operated 'battery', and around it are 'resistors', 'switches', 'junctions', a 'capacitor', and a 'gramophone horn' which plays sounds when it is turned, and works like an 'ammeter'. Act 2 adds transistors and an inductor

STRING POWER

Rather than electricity, you have a drawstring-charged spring 'battery' which drives the output sprocket for a surprisingly long time when you pull the string out and release it. Tiny chains take the place of wires transferring power to components, including rotary resistors and switches. Higher values of resistance are harder to turn, and you can use the push-button switches to stop and start a segment of chain moving. All the components have magnetic bases that hold them firmly on the metal baseplates.

Lovely touches abound. The spring-powered battery has a 'capacity' (the amount of energy you get from pulling the string all the way out), and the time it takes to deplete the battery depends on how much power you take from it. Use it to turn a high-value resistor, and it runs for a while. Send the power through multiple

> The use of chain as an analogue for wire is **mostly intuitive, but not without its issues**

signal paths into several resistors, and the string will be drawn back into the battery a lot more quickly. The battery itself has an 'overload' detector that stops it turning if you allow it to run with nothing connected to the output.

The use of chain as an analogue for wire is mostly intuitive, but not without its issues. However, once you understand how it works, you can start to convert electrical circuits to mechanical ones. If you want to split the current flow and connect components in parallel, you use one of the blue junction devices which contain a differential gear. That does a splendid job of showing both how current flows in a circuit with parallel wired components, and how mechanical power can be shared out.

Once you've handled a transistor component and seen how it functions, you can build a stronger understanding of what the electronic device does. There are some fiddly moments to be had when building the circuits though. Getting all the chains tight can be tricky. And the length of chain supplied is only just enough to make some of the larger circuits.

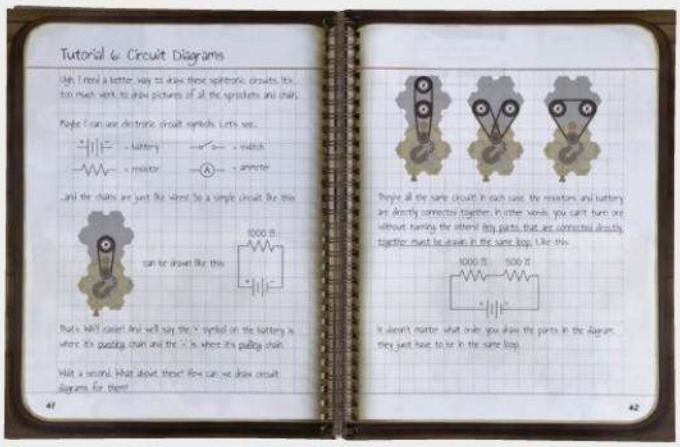
The sets come with beautifully illustrated puzzle books that tell the story of Natalia and her clock-making family, and provide plenty of exercises. The books are well-printed and spiral-bound, so that they lay flat on the desk. The Act 1 set provides a good starting point, but for the interesting stuff (including the amazing inductor flywheel component), you really need Act 2.



Above
The circuit above shows how we can use Spintronics to make a mechanical circuit containing a switched resistor. The switch is on the left. Press the button to turn it off and it locks the sprocket, preventing movement. Press again to turn on and current starts to flow as the chain moves. The pitch of the sound produced by the 'ammeter' horn changes depending on the value of resistor at the top of the circuit. The pullstring at the bottom of the picture has almost been drawn all the way back into the battery

If you fancy making your own components, you can buy a Virtual Pack of the kit designs as a downloadable file. The designs would be a bit hard to print on a home 3D printer, but they make it a lot easier to get started with your own custom components. The sets are not cheap, but each is not much more than a full-priced video game, and they are very well made and presented. They introduce a mapping of physical to electrical thinking that really helps you understand the underlying principles behind both.

Overall, this is a fun and original introduction to electronics. ◻



Left
The tutorials use electrical notation in all of the examples, and each mechanical component is mapped closely to an electrical one

VERDICT

The whole thing is a labour of love. The attention to detail in the mouldings is amazing, and everything just works.

9/10

REVIEW

Raspberry Pi Debug Probe

Peek inside the mind of your microcontroller

RASPBERRY PI LTD ◆ **$12** | **hsmag.cc/debugprobe**

By Ben Everard @ben_everard

Debuggers let you see into the heart of your program. You can watch it execute line by line and watch it change the variables, or leave it running to a particular point, then make sure everything is correct before kicking it off even further. They can do this because they're wired into a special bit of control hardware in the processor.

The downside of this is that you need an extra bit of hardware to tap into this.

Raspberry Pi Pico – like most Arm-based microcontrollers – has a Serial Wire Debug (SWD) port which gives you access to the debug controller in the processor. To use this, you need something that can communicate via SWD. On a Raspberry Pi, you can use the GPIO pins, but most other computers don't have this option. The Raspberry Pi Debug Probe provides a link between your computer and the SWD port on Pico.



Below ◆
The plastic case will keep the Probe safe in a messy box of components

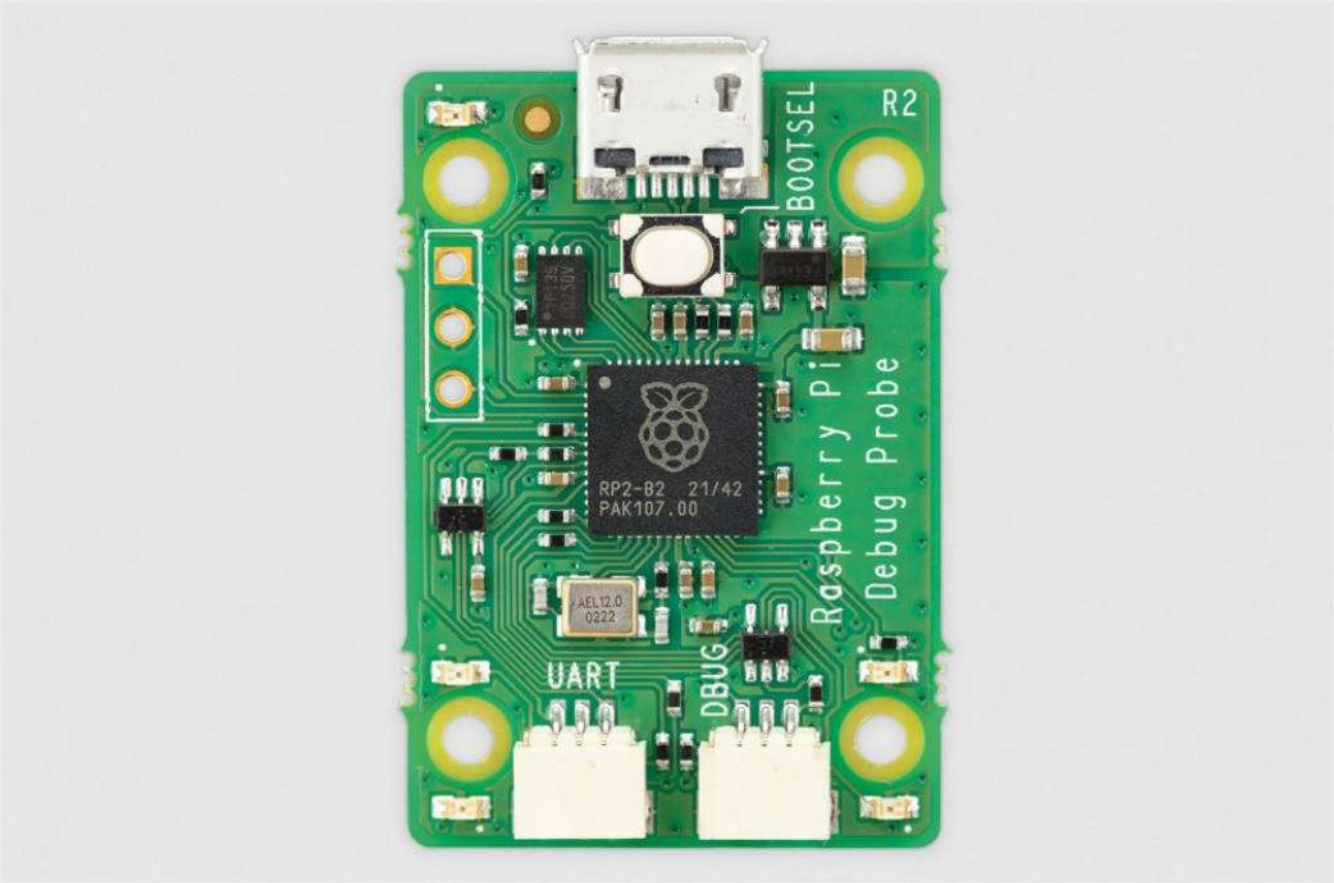
You'll also need some software running on your computer that can communicate with the Debug Probe to get it to do what you want. This is usually a combination of OpenOCD and GDB. You can run these from the command line or through a development environment, such as Visual Studio Code.

There are a lot of capabilities, but the most common ones are setting breakpoints, which are places in code where the program will pause and let you examine what's going on; stepping through code line by line to see what happens; and examining how variables change as you do this. This gives you a much more detailed picture of what's going on inside the tiny silicon brain of your Pico than you can easily get with any other method.

The Raspberry Pi Debug Probe also has a UART connection which can be useful for connecting to other hardware, or working with programs that don't use the debug capabilities.

Of course, there's nothing new about debuggers, and there is other hardware out there that supports SWD and UART. In fact, you can even use a regular Pico as a debug probe using the picoprobe firmware.

There are a few advantages of the Raspberry Pi Debug Probe over other debuggers. Perhaps the biggest one is that it connects easily if you're using Pico H or Pico WH. These two models come with headers attached, and have slightly different connectors for SWD. You can get an adaptor cable to



connect these to other debuggers, but the Raspberry Pi debugger can just plug straight in.

If you're not using an 'H' version, then there are male and female header connectors that you can wire up.

As well as these cables, the Debug Probe comes with a sturdy plastic case and a micro USB cable. While the micro USB cable might be a controversial choice for some, it does mean that the same cable works for both Pico and Debug Probe. It's also in a distinctive Raspberry colour, which means it's easy to spot amongst a sea of black or white power-only cables.

Debuggers are one of those tools that, when they work properly, are very straightforward. With the Raspberry Pi Debug Probe, it all started working as expected when we followed the documentation.

WHICH LANGUAGE?

Not all programming languages support debug hardware. For instance, neither MicroPython nor CircuitPython do. Debug probes are mostly used with C and C++. When developing in either of these languages, it is frankly far easier, particularly as your projects get more complex. Yes, you can just throw in hundreds in printfs, and you'll likely manage to find the problem eventually, but you'll probably get to the heart of the bug much faster with a debug probe.

> **The Raspberry Pi Debug Probe brings** it all into a sturdy bit of hardware

While any debug probe gets you all this, the Raspberry Pi Debug Probe brings it all into a sturdy bit of hardware that connects easily to any model of Pico, and is well covered by the Pico documentation. For $12, you'll very quickly save a lot of time chasing bugs around your code.

Above
The Debug Probe is powered by the same RP2040 chip that's at the heart of the Pico

VERDICT

If you're programming C or C++ on Pico, this is probably the best $12 you can spend.

10/10

CROWDFUNDING NOW

Wind turbines

Renewable energy for when the sun doesn't shine

From £18 | hsmag.cc/windpower | Delivery: from April 2023

Walk through any town in the UK, and you'll see shiny solar panels on the roofs of many houses. These are a great part of a sustainable future, but they're only part of it. After all, the sun doesn't shine every day of the year. In fact, it sometimes feels like it rarely shines.

If we're going to move into a future where we can still have our creature comforts, we will need more ways of harvesting energy from the environment, and the obvious second option is wind power. There are huge offshore farms already, but why don't we have smaller units that we can deploy in a domestic setting?

There might be a very good reason that they're not deployed more in a domestic setting but, as far as we've been able to tell, it's not been widely explored. This is why we're excited by the innovative horizontal-vertical wind turbine systems that Ali El Zaher Buendia has launched on Kickstarter.

There are three designs between 15 and 400 W, made from 3D-printed components and a few off-the-shelf parts. The reward levels start with just the STL files (for £18) and go up to the 400 W system (for £150). In all cases, you'll need to supply a few additional bits of hardware, including bearings and a motor that's used as a generator.

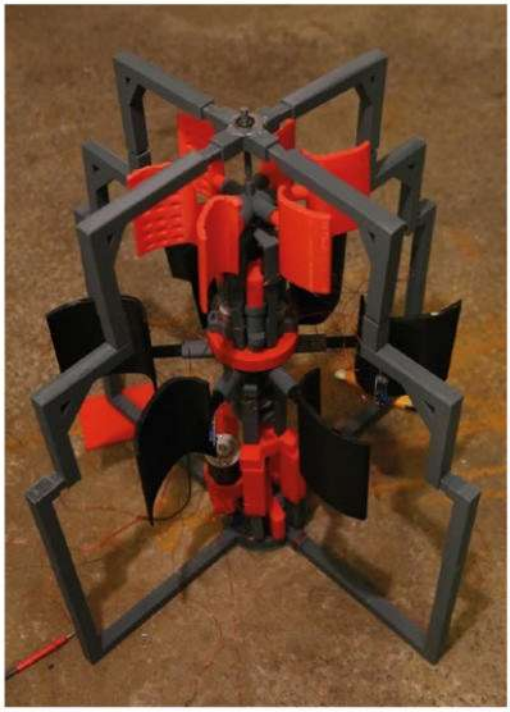
We haven't tested this build out, so can't say what sort of power you're likely to get out of it in real-world usage, or how reliable it is, but at the very least, it does look like a cost-efficient, and not too difficult, way of getting started with small-scale wind power generation which has, so far, not been an option.



Left The single-stage setup can produce up to 100 watts of power

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.



Right
You can add a second set of blades on top, to provide more power

Below
The turbine is small enough to take with you on your adventures

> **You'll need to supply a few additional bits of hardware,** including bearings and a motor



next month

issue #66

ON SALE
20 APRIL

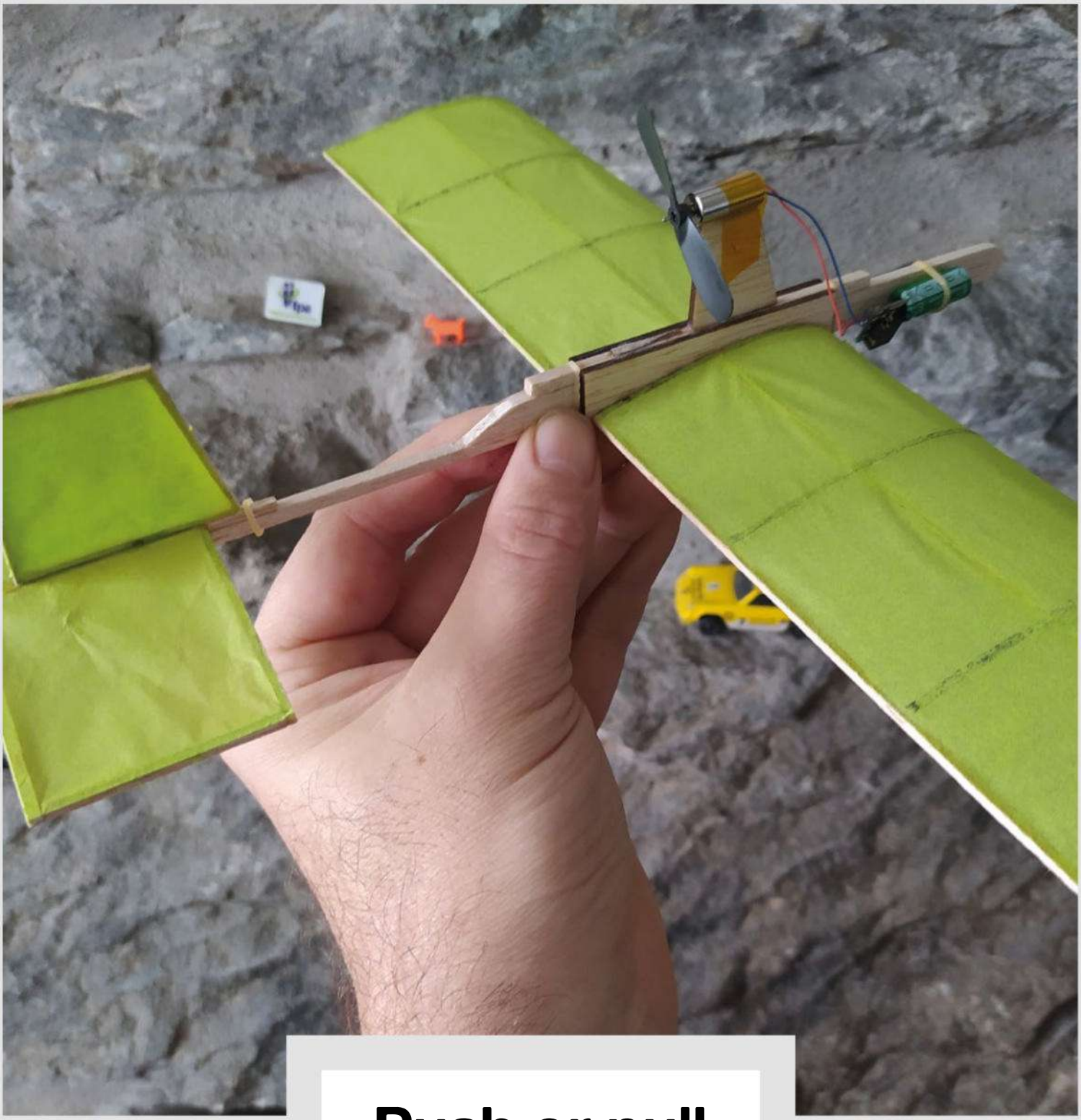
3D PRINTED AUTOMATA

ALSO

- **PCB DESIGN**
- **3D PRINTING**
- **CIRCUITPYTHON**
- **LASERS**
- **AND MUCH MORE**

DON'T MISS OUT

hsmag.cc/subscribe



Push or pull

Nowadays, when most people think of aeroplanes, they think of a propeller at the front, but this wasn't always the case. When the Wright brothers built their first plane, they put the propeller behind the wings. This 'pusher' design was popular for a few years before the now familiar 'tractor propeller' design took over.

While full-scale planes might have abandoned pushers, they're still great for small-scale designs. Keeping the propeller tucked behind the wings and the fuselage keeps it safe (or at least safer) in crashes. This might not be a significant concern for commercial airlines, but it makes it easier to keep flying our models.

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects



Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io